С. М. Устинов, В. А. Зимницкий

ВЫЧИСЛИТЕЛЬНАЯ МАТЕМАТИКА

- *Аппроксимация функций и смежные вопросы*
- *Задачи линейной алгебры*
- *Нелинейные уравнения и системы*
- *Методы решения дифференциальных уравнений*
- *Введение в минимизацию функций*

УЧЕБНОЕ ПОСОБИЕ



С. М. Устинов
В. А. Зимницкий

ВЫЧИСЛИТЕЛЬНАЯ МАТЕМАТИКА

Рекомендовано Учебно-методическим объединением по университетскому политехническому образованию в качестве учебного пособия для студентов высших учебных заведений, обучающихся по направлениям подготовки 220100 «Системный анализ и управление» и 230100 «Информатика и вычислительная техника»

Санкт-Петербург
«БХВ-Петербург»
2009

УДК 681.3.06(075.8)
ББК 32.973.26-018.2я73
У80

Устинов, С. М.

У80 Вычислительная математика / С. М. Устинов, Зимницкий В. А. — СПб.: БХВ-Петербург, 2009. — 336 с.: ил. — (Учебное пособие)

ISBN 978-5-9775-0318-1

Изложены аппроксимация функций и смежные вопросы, задачи линейной алгебры, нелинейные уравнения и системы, методы решения дифференциальных уравнений, введение в минимизацию функций. Особое внимание обращается на реальные трудности, возникающие на практике при аппроксимации и минимизации функций, при решении этих задач. Важное место в изложении материала занимают проблема плохой обусловленности при решении линейных систем алгебраических уравнений, явление жесткости в дифференциальных уравнениях и явление овражности при минимизации функций. Дается представление о том, как строится программное обеспечение для обсуждаемых методов.

Для студентов, аспирантов, преподавателей технических вузов и инженеров

УДК 681.3.06(075.8)
ББК 32.973.26-018.2я73

Рецензенты:

Козлов В. Н., д. т. н., профессор, проректор Санкт-Петербургского государственного политехнического университета (СПбГПУ)
Александров А. М., д. т. н., профессор, зам. начальника Центра анализа и экспертизы ФГУП НПО «Импульс»

Группа подготовки издания:

Главный редактор	*Екатерина Кондукова*
Зам. главного редактора	*Татьяна Лапина*
Зав. редакцией	*Григорий Добин*
Редактор	*Анна Кузьмина*
Компьютерная верстка	*Натальи Смирновой*
Корректор	*Виктория Пиотровская*
Оформление обложки	*Елены Беляевой*
Зав. производством	*Николай Тверских*

Лицензия ИД № 02429 от 24.07.00. Подписано в печать 28.08.08.
Формат $70\times100^1/_{16}$. Печать офсетная. Усл. печ. л. 27,09.
Тираж 2000 экз. Заказ № 3418
"БХВ-Петербург", 194354, Санкт-Петербург, ул. Есенина, 5Б.

Санитарно-эпидемиологическое заключение на продукцию № 77.99.60.953.Д.003650.04.08 от 14.04.2008 г. выдано Федеральной службой по надзору в сфере защиты прав потребителей и благополучия человека.

Отпечатано с готовых диапозитивов
в ГУП "Типография "Наука"
199034, Санкт-Петербург, 9 линия, 12

ISBN 978-5-9775-0318-1

Оглавление

Введение

Это учебное пособие написано на базе лекций, читавшихся на факультете технической кибернетики Санкт-Петербургского государственного политехнического университета. Их основы более тридцати лет назад были заложены профессором Ю. В. Ракитским, а затем многократно перерабатывались и расширялись авторами.

В настоящее время имеется достаточное количество учебников с названием "Численные методы" или похожим на него. В первую очередь, следует отметить книгу Н. С. Бахвалова и др. [1], построенную на основе лекций, читавшихся на механико-математическом факультете и факультете вычислительной математики и кибернетики МГУ. Этот весьма фундаментальный и обстоятельный курс характеризуется большой продолжительностью и сочетает необходимую строгость изложения с обсуждением многих реально возникающих на практике проблем. В качестве предыдущей версии университетского курса следует упомянуть книгу И. С. Березина и Н. П. Жидкова [2, 3]. Имеется также большое число книг, в основном, для студентов инженерных специальностей, где авторы приводят многочисленные методы, к сожалению, без должного обоснования, ориентируя читателей на использование стандартных программ.

В предлагаемом учебном пособии, предназначенном для студентов и аспирантов технических вузов, авторы пытались достичь определенного компромисса между этими двумя подходами и одновременно преследовали следующие цели. Во-первых, везде, где возможно, сохранить разумную строгость изложения, указать способ построения того или иного метода и сократить ситуации, когда алгоритмы решения задачи приводятся без вывода. Во-вторых, максимально полно познакомить читателя с основными трудностями, возникающими на практике при решении обсуждаемой проблемы. Так, например, важное место в пособии занимают проблема плохой обусловленности при решении линейных систем алгебраических уравнений, явление жесткости в дифференциальных уравнениях и явление овражности при минимизации функций, имеющие в своей основе немало общего и заставляющие расплачиваться либо точностью решения, либо большим объемом вычислений. В-третьих, дать первое представление о том, как строится

программное обеспечение для обсуждаемых методов. За основу и в качестве примера были взяты программы из книги Дж. Форсайта, М. Малькольма, К. Моулера "Машинные методы математических вычислений" [14]. И, хотя на русском языке был издан ее переработанный и расширенный вариант [7], без ущерба для понимания основных проблем предпочтение было отдано [14] в связи с публикацией полного текста всех используемых программ непосредственно в книге. В-четвертых, ставилась цель построить изложение так, чтобы оно отвечало курсу объемом порядка тридцати лекций.

Перечисленные цели во многом противоречили друг другу, и достигнутый компромисс целиком определялся субъективными взглядами авторов. Так, например, материал, посвященный методам решения систем линейных алгебраических уравнений и уравнений в частных производных, следует считать лишь введением в эти разделы. Более подробно они изложены в учебниках, приведенных в списке литературы. Оговоренное количество лекций не позволило отразить такие безусловно важные проблемы, как работа с разреженными матрицами и параллельные вычисления. Для самостоятельного ознакомления с ними можно рекомендовать книги [38, 26, 55, 21, 35] и др.

Для сокращения объема учебного пособия часто не делалось специальных оговорок в отношении свойств используемых функций. Например, если какая-то функция дифференцируется или интегрируется, то по умолчанию предполагается, что она обладает всеми необходимыми свойствами для выполнения этих операций.

В тексте пособия содержится некоторое количество вопросов для читателя. Об ответах на них при первом чтении можно не задумываться. Однако более серьезная работа над лекциями заставит вернуться к ответам для уточнения и закрепления полученных знаний. В ряде случаев вопросы носят откровенно провокационный и даже математически некорректный характер и не имеют однозначного ответа. Здесь основная цель — заставить студента задуматься и самостоятельно осмыслить обсуждаемые проблемы.

Предполагается, что читатель знаком с курсом высшей математики в традиционном объеме для технического вуза, а также с материалами, размещенными в приложениях. При самостоятельном изучении пособия мы рекомендуем начинать с материалов первых четырех приложений. Когда пособие используется при чтении лекций, сведения из приложений можно излагать частями по мере возникновении в них потребности в остальных главах.

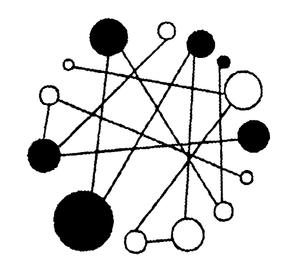
В *приложении 5* отражены примеры упражнений, лабораторных и курсовых работ, предусмотренных учебными планами. Традиционно такая проблема-

тика требует самостоятельных методических указаний большого объема с детальной проработкой. В данном случае авторы пытались лишь проиллюстрировать возможное наполнение практических занятий и тем самым помочь студенту, самостоятельно изучающему пособие, а также молодому преподавателю.

Приводимый список литературы существенно ограничен и не претендует на полноту. С одной стороны, это перечень "универсальных" учебников, ориентированных на разнообразные численные методы широкого круга задач [1—15]. С другой стороны, у вдумчивого и пытливого читателя часто возникает вопрос: "А что еще можно почитать на эту тему?" Для этих студентов мы и делаем ссылки, также не претендующие на полноту:

- глава 2 — [48, 46, 19, 20, 25, 29, 30, 36, 39, 47, 49];
- глава 3 — [34, 27];
- глава 4 — [50, 51, 16, 33, 40, 42, 43, 52];
- глава 5 — [18, 24, 27, 37, 40, 54];
- глава 6 — [32, 40, 45];
- приложения — [23, 22, 46, 17, 31, 44, 53].

Глава 1



Аппроксимация функций и смежные вопросы

1.1. Общие сведения

Этот раздел посвящен некоторым аспектам теории приближения функций. Термин "аппроксимация" (от лат. *approximare* — приближаться) в данном контексте трактуется как "замена".

Прежде чем излагать какие-либо подходы к решению этой проблемы, целесообразно рассмотреть следующие вопросы. *Что нужно заменять и зачем заменять? Чем заменять? Как количественно оценивать погрешность замены? Наконец, если есть несколько вариантов замены, то как выбрать из них наилучший?*

Итак, что заменять. На математическом языке это означает замену одной функциональной зависимости другой. Исходная функция чаще всего задается в одном из следующих видов:

- аналитически;
- графически;
- таблично;
- алгоритмически.

В последнем случае подразумевается, что аналитический вид функции неизвестен, но задан алгоритм (быть может, весьма сложный и трудоемкий), ставящий в соответствие любому значению аргумента x из области определения значение функции $f(x)$. Так как на практике часто возникает потребность дифференцировать, интегрировать $f(x)$ или использовать ее в различных расчетах, то целесообразность замены $f(x)$ другой функцией, имеющей ана-

литический вид, сомнений не вызывает. Даже если $f(x)$ задана аналитически, она может оказаться весьма сложной с позиций решаемой задачи, и требуется другая заменяющая ее функция. При выборе аппроксимирующей функции обычно руководствуются ее простым видом.

Теоретическим обоснованием выбора заменяющей функции может служить *теорема Вейерштрасса*: если $f(x)$ — произвольная непрерывная на конечном замкнутом интервале $[a, b]$ функция, то для любого $\varepsilon > 0$ найдется такой полином $P_n(x)$ степени $n = n(\varepsilon)$, что

$$\max\left|f(x) - P_n(x)\right| < \varepsilon .$$

Однако эту теорему следует отнести к чистым теоремам существования. Она не дает гарантии, что такой полином можно построить с помощью практичного алгоритма. Тем не менее, именно полиномы являются наиболее популярными заменяющими функциями, поскольку они удобны в работе и их свойства хорошо известны. Наряду с ними используют тригонометрические, экспоненциальные функции и т. д. В общем случае требования к простоте выдвигает решаемая задача.

Для того чтобы можно было сравнивать различные варианты аппроксимации, следует ввести критерий близости. В частности, им может быть максимум модуля отклонения исходной функции $f(x)$ от аппроксимирующей $g(x)$ на заданном промежутке

$$\delta = \max_{x\in[a,b]}\left|f(x) - g(x)\right| \qquad (1.1.1)$$

или так называемый "среднеквадратичный критерий"

$$\rho^2 = \int_a^b \left(f(x) - g(x)\right)^2 dx. \qquad (1.1.2)$$

В том случае, когда $f(x)$ определена таблично на заданном наборе точек, может быть использован дискретный аналог критерия (1.1.2):

$$\rho^2 = \sum_{k=1}^{N}\left(f(x_k) - g(x_k)\right)^2. \qquad (1.1.3)$$

Лучшей оказывается аппроксимирующая функция, обладающая наименьшей величиной δ или ρ^2. Для понимания дальнейшего полезно ответить на следующие вопросы.

Вопрос 1

Не являются ли критерии (1.1.1) и (1.1.2) весьма похожими, и если аппроксимирующая функция $g1(x)$ лучше, чем $g2(x)$ по одному из критериев, то окажется ли она лучше и по другому?

Легко заметить, что вопрос носит провокационный характер, о чем свидетельствует пример на рис. 1.1, когда лучшими по каждому критерию оказываются различные функции.

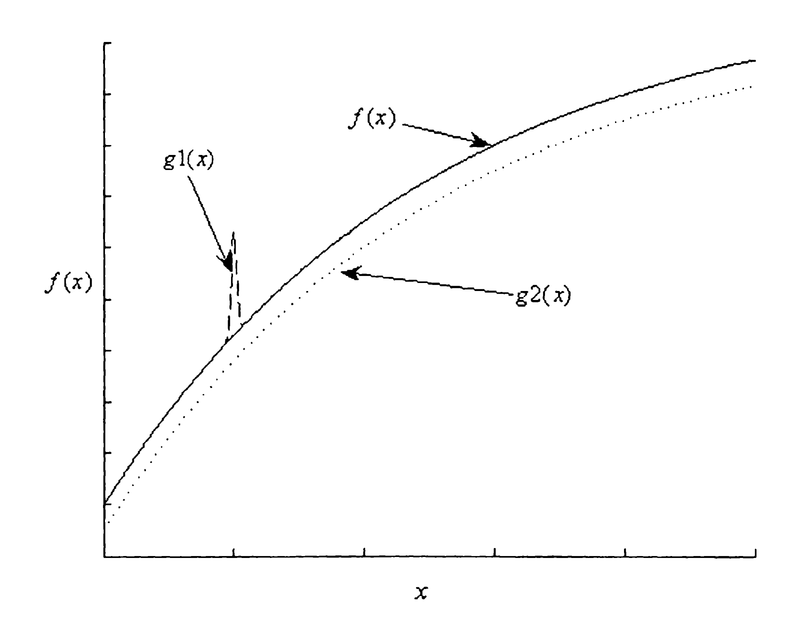


Рис. 1.1. Различные варианты аппроксимации

Вопрос 2

На практике какая из аппроксимирующих функций на рис. 1.1 предпочтительнее, $g1(x)$ или $g2(x)$?

И опять вопрос носит провокационный характер, т. к. словосочетание "на практике" еще не задает критерий. Только решаемая задача диктует выбираемый критерий близости, который, в свою очередь, позволяет выбрать лучшую аппроксимацию.

Теперь приступим к подробному изложению задачи интерполяции, основанной на интерполяционном критерии близости.

1.2. Постановка задачи интерполирования

Будем приближать исходную функцию, заданную таблично:

x	x_0	x_1	x_2	...	x_m
$f(x)$	$f(x_0)$	$f(x_1)$	$f(x_2)$	...	$f(x_m)$

обобщенным многочленом

$$Q_m(x) = a_0\varphi_0(x) + a_1\varphi_1(x) + \ldots + a_m\varphi_m(x) = \sum_{k=0}^{m} a_k\varphi_k(x), \tag{1.2.1}$$

где $\{\varphi_k\}$ — заданный набор линейно независимых функций, а коэффициенты a_k подлежат определению. При этом в качестве критерия близости выбирается совпадение значений $f(x)$ и $Q_m(x)$ в узлах таблицы

$$Q_m(x_i) = f(x_i), \quad i = 0,\ 1,\ \ldots,\ m. \tag{1.2.2}$$

Тогда $Q_m(x)$ называется *интерполяционным многочленом*, а x_k — *узлами интерполирования*.

Равенства (1.2.2) представляют собой систему линейных алгебраических уравнений относительно искомых коэффициентов обобщенного многочлена a_0, a_1, ..., a_m. Эта система имеет единственное решение, если ее определитель отличен от нуля:

$$\det\begin{pmatrix} \varphi_0(x_0) & \varphi_1(x_0) & \ldots & \varphi_m(x_0) \\ \varphi_0(x_1) & \varphi_1(x_1) & \ldots & \varphi_m(x_1) \\ \ldots & \ldots & \ldots & \ldots \\ \varphi_0(x_m) & \varphi_1(x_m) & \ldots & \varphi_m(x_m) \end{pmatrix} \neq 0.$$

Последний факт имеет место при любом упорядоченном наборе неповторяющихся узлов для системы линейно независимых функций $\{\varphi_k(x)\}$, если

все эти функции $m+1$ раз дифференцируемы на промежутке, где расположены узлы, и все определители Вронского

$$W\left(\varphi_0(x), \varphi_1(x), \ldots, \varphi_k(x)\right) = \begin{pmatrix} \varphi_0(x) & \varphi_1(x) & \ldots & \varphi_k(x) \\ \varphi_0'(x) & \varphi_1'(x) & \ldots & \varphi_k'(x) \\ \ldots & \ldots & \ldots & \ldots \\ \varphi_0^{(k)}(x) & \varphi_1^{(k)}(x) & \ldots & \varphi_k^{(k)}(x) \end{pmatrix}$$

отличны от нуля на $[a, b]$.

Наиболее популярной является полиномиальная аппроксимация, когда $\varphi_k(x) = x^k$, $Q_m(x) = a_0 + a_1 x + a_2 x^2 + \ldots + a_m x^m$, а определитель системы (1.2.2) приобретает вид

$$\begin{vmatrix} 1 & x_0 & x_0^2 & . & x_0^n \\ 1 & x_1 & x_1^2 & . & x_1^n \\ 1 & x_2 & x_2^2 & . & x_2^n \\ . & . & . & . & . \\ 1 & x_n & x_n^2 & . & x_n^n \end{vmatrix}. \qquad (1.2.3)$$

Этот определитель называется *определителем Вандермонда*. Он отличен от нуля, и задача имеет единственное решение, если узлы интерполирования x_0, x_1, ..., x_m различны.

1.3. Интерполяционный полином Лагранжа. Остаточный член полинома Лагранжа

Непосредственное численное решение линейной системы (1.2.2) представляет значительные трудности. С одной стороны, это связано с заметным объемом вычислений для нахождения a_k. С другой стороны, малое изменение данных таблицы (x_k, $f(x_k)$) часто приводит к сильному изменению решения (особенно для близко расположенных узлов интерполирования). В связи с этим, естественно попытаться построить интерполяционный полином, не решая системы (1.2.2).

В качестве базиса вместо $\varphi_k(x) = x^k$ выберем полиномы $\omega_0(x)$, $\omega_1(x)$, ..., $\omega_m(x)$ такие, что они обращаются в нуль во всех узлах, кроме одного, когда индекс полинома совпадает с индексом узла:

$$\omega(x) = (x - x_0)(x - x_1)...(x - x_m), \tag{1.3.1}$$

$$\omega_k(x) = \frac{\omega(x)}{(x - x_k)} = (x - x_0)(x - x_1)...(x - x_{k-1})(x - x_{k+1})...(x - x_m). \tag{1.3.2}$$

В этих обозначениях запишем следующий полином:

$$Q_m(x) = \sum_{k=0}^{m} \frac{\omega_k(x)}{\omega_k(x_k)} f(x_k). \tag{1.3.3}$$

По построению это многочлен степени m. Определим его значения в узлах интерполирования x_i. Так как для $x = x_i$ полином $\omega_k(x)$ равен нулю, если только $i \neq k$, то для $Q_m(x_i)$ получаем

$$Q_m(x_i) = \sum_{k=0}^{m} \frac{\omega_k(x_i)}{\omega_k(x_k)} f(x_k) = \frac{\omega_i(x_i)}{\omega_i(x_i)} f(x_i) = f(x_i), \quad i = 0,\ 1,\ ...,\ m.$$

Таким образом, $Q_m(x)$ — интерполяционный полином, получивший название *интерполяционный полином Лагранжа.*

Вопрос 3

По заданной таблице при отсутствии ошибок округления построили два полинома. Коэффициенты первого нашли решением системы (1.2.2). Второй был непосредственно воспроизведен по формуле (1.3.3). Какой из этих полиномов лучше приближает исходную функцию?

Интерполяционные многочлены можно строить и по другим системам базисных функций, например, таким как

$$1,\ \cos x,\ \sin x,\ \cos 2x,\ \sin 2x,\ ...,\ \cos nx,\ \sin nx,\ ...$$

$$1,\ e^{\alpha_1 x},\ e^{\alpha_2 x},\ ...,\ e^{\alpha_n x},\ ...$$

Теперь обратимся к погрешности интерполяционного полинома. Исходная функция $f(x)$ может быть представлена в виде

$$f(x) = Q_m(x) + R_m(x), \tag{1.3.4}$$

где $Q_m(x)$ — интерполяционный полином, а $R_m(x)$ носит название *остаточного члена интерполяционного полинома.* Вид $R_m(x)$ получается на основе известной теоремы Ролля, которая утверждает следующее.

Если $f(x)$ непрерывна на отрезке $[a, b]$, имеет первую производную в каждой точке внутри этого отрезка, и значения функции на концах этого промежутка равны, т. е. $f(a) = f(b)$, то внутри отрезка найдется, по крайней мере, одна такая точка $x = c$, что $f'(c) = 0$.

Теорема. Если $f(x)$ на промежутке $[a, b]$ имеет непрерывные производные вплоть до $m+1$ порядка, то остаточный член $R_m(x)$ можно представить в виде:

$$R_m(x) = f(x) - Q_m(x) = \frac{f^{(m+1)}(\eta)}{(m+1)!}\omega(x), \quad \eta \in [a, b]. \tag{1.3.5}$$

При этом $\omega(x)$, как и прежде, определяется формулой (1.3.1).

Доказательство. Рассмотрим вспомогательную функцию

$$\varphi(z) = f(z) - Q_m(z) - K\omega(z), \tag{1.3.6}$$

где K — некоторая постоянная. Пусть x_k — узлы интерполирования, а x — точка, в которой оценивается погрешность $(x \neq x_k)$. Легко заметить, что функция $\varphi(z)$ равна нулю во всех узлах интерполирования. Выберем константу K так, чтобы $\varphi(x) = 0$.

$$K = \frac{f(x) - Q_m(x)}{\omega(x)} = \frac{R_m(x)}{\omega(x)}. \tag{1.3.7}$$

Таким образом, $\varphi(z)$ имеет по меньшей мере $m+2$ нуля (все узлы интерполирования и точка x). Тогда по теореме Ролля первая производная $\varphi(z)$ имеет по меньшей мере $m+1$ нуль, вторая производная — не менее m нулей, а $(m+1)$-я производная $\varphi^{(m+1)}(z)$ имеет по меньшей мере один нуль. Обозначим точку, где $(m+1)$-я производная обращается в нуль за η. Тогда, последовательно дифференцируя (1.3.6), получаем

$$\varphi^{(m+1)}(\eta) = f^{(m+1)}(\eta) - 0 - K(m+1)! = 0.$$

Подставляя в это равенство выражение (1.3.7) для K, получаем формулу для $R_m(x)$, совпадающую с (1.3.5).

Формула (1.3.5) позволяет сделать очевидный, но важный вывод. Пусть $f(x)$ — это полином степени m. Тогда $f^{(m+1)}(\eta)=0$. Следовательно, полином степени m *однозначно* воспроизводится интерполяционным полиномом по $(m+1)$ точке.

Ясно также, что остаточный член во всех узлах интерполирования равен нулю.

В заключение отметим тот факт, что, хотя о расположении точки η на промежутке интерполирования ничего не известно, очевидна зависимость величины η как от узлов интерполирования, так и от точки x, где оценивается погрешность, т. е. $\eta=\eta(x)$.

Остаточный член позволяет оценивать отклонение $L_m(x)$ от $f(x)$ для дифференцируемых функций тогда, когда удается оценить $f^{(m+1)}(x)$. Полагая $M_{m+1}=\max\left|f^{(m+1)}(x)\right|$, получим $\left|R_m(x)\right| \le \frac{M_{m+1}}{(m+1)!}\left|\omega(x)\right|$.

1.4. Выбор узлов интерполирования

Для уменьшения погрешности интерполирования обратимся к формуле (1.3.5) при заданной степени полинома m. Поскольку величиной $f^{(m+1)}(\eta)$ трудно управлять, и возможна лишь оценка пределов ее изменения, задача уменьшения погрешности сводится к управлению величиной $\left|\omega(x)\right|$ за счет выбора узлов интерполирования. Рассмотрим два типичных на практике случая.

Случай 1. Задана степень полинома m, и имеется таблица достаточно большой длины. Точка x^*, в которой вычисляется значение полинома, заранее известна. Требуется выбрать $m+1$ узел так, чтобы величина $\left|\omega(x^*)\right|$ была минимальной. Результат очевиден. Нужно выбирать узлы интерполирования из таблицы, *ближайшие* к x^*. Использование любого другого узла вместо ближайшего неизбежно увеличивает значение

$$\left|\omega(x^*)\right|=\left|\left(x^*-x_0\right)\left(x^*-x_1\right)\ldots\left(x^*-x_m\right)\right|.$$

Случай 2. Заданы степень полинома m и промежуток интерполирования $[a, b]$. Точка x^*, в которой вычисляется значение полинома, заранее не известна. Требуется выбрать узлы интерполирования так, чтобы в самом неблагоприятном случае расположения x^* погрешность была минимальной (так называемый *минимаксный критерий*):

$$\max_{x^* \in [a, b]} \left|\omega(x^*)\right| \to \min. \qquad (1.4.1)$$

Вопрос 4

С учетом того, что величина x^* заранее неизвестна и нет оснований отдавать предпочтение какой-либо части промежутка, как следует задавать узлы x_k на $[a, b]$?

Интуитивно часто напрашивающееся предложение о равномерном задании узлов на промежутке оказывается ошибочным. Так как значения $\left|\omega(x^*)\right|$ в узлах интерполирования равны нулю, график $\left|\omega(x^*)\right|$ напоминает "колокольчики", максимум которых достигается между узлами интерполирования. Одинакова ли высота этих "колокольчиков" на различных промежутках между узлами? Оказывается, что нет. Выбор узлов равноотстоящими с попыткой уравновесить погрешность на отдельных участках промежутка $[a, b]$ оказывается здесь неудачным, и точки, расположенные ближе к центру промежутка, попадают в привилегированное положение. Поясним это на примере.

Расположим на промежутке $[0, 5]$ равномерно 6 узлов с шагом $h = 1$ ($x_k = k$, $k = 0, 1, 2, 3, 4, 5$).

Для точки $x^* = 2.5$ такие узлы, как это рассматривалось в случае 1, оказываются наилучшими при единичном шаге $\left(\left|\omega(2.5)\right| = 2.5 \times 1.5 \times 0.5 \times 0.5 \times 1.5 \times 2.5 \approx 3.52\right)$. В то же время для $x^* = 0.5$ получаем $\left|\omega(0.5)\right| = 0.5 \times 0.5 \times 1.5 \times 2.5 \times 3.5 \times 4.5 \approx 14.77$. Здесь в соответствии с рекомендациями к случаю 1 вместо узлов $x_4 = 4$ и $x_5 = 5$ следовало бы взять ближайшие узлы $x_4 = -1$ и $x_5 = -2$, но они в таблице отсутствуют. График $\left|\omega(x)\right|$, представленный на рис. 1.2, не только отражает сказанное, но подсказывает выход из создавшейся ситуации. Узлы интерполирования нужно симметрично сместить ближе к концам промежутка. Тогда высота центрального

"колокольчика" увеличится, в то время как высота крайних уменьшится. Оптимальный выбор узлов интерполирования отвечает нулям так называемых ортогональных полиномов Чебышева (о них речь пойдет позже), когда все "колокольчики" будут одинаковыми по высоте. Соответствующий график представлен на рис. 1.3, а значение x_k определяются по формулам

$$x_i = \frac{b+a}{2} + \frac{b-a}{2}\xi_i, \text{ где } \xi_i = -\cos\frac{2i+1}{2m+2}\pi, \quad i = 0...m. \tag{1.4.2}$$

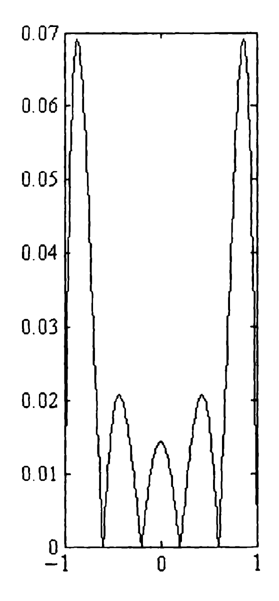


Рис. 1.2. Равноотстоящие узлы

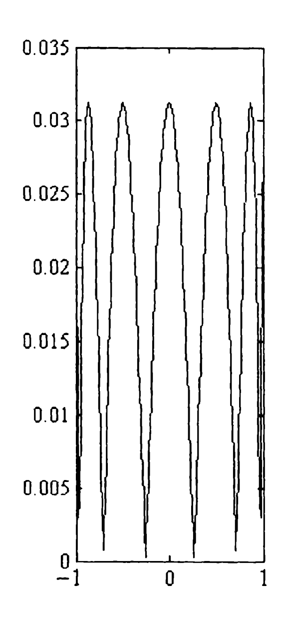


Рис. 1.3. Чебышевские узлы

1.5. Интерполяционный полином Ньютона для равно- и неравноотстоящих узлов

Оценка погрешности непосредственно на основе формулы (1.3.5) выполняется крайне редко из-за известных трудностей, связанных с оценкой производной $f^{(m+1)}(\eta)$, особенно для таблично заданной функции. Поэтому на практике о величине погрешности принято судить, сравнивая в заданной точке x^* значе-

ния интерполяционных полиномов соседних степеней $Q_m\left(x^*\right)$ и $Q_{m+1}\left(x^*\right)$. При недостаточной точности последовательно повышают степень полинома. Но для такой процедуры использование полинома Лагранжа (1.3.3) оказывается неэффективным. При переходе от полинома $Q_m(x)$ к полиному $Q_{m+1}(x)$ почти всю работу приходится выполнять заново. Действительно, добавление еще одного узла интерполирования связано не только с появлением дополнительного слагаемого в формуле (1.3.3), но и с перерасчетом всех остальных слагаемых в (1.3.3) из-за возникновения x_{m+1}. Целесообразно записать полином $Q_{m+1}(x)$ в таком виде, чтобы расчеты сводились к появлению лишь еще одного слагаемого в дополнение к ранее вычисленному $Q_m(x)$. С этой целью запишем первую разделенную разность *(см. разд. П1.2)*:

$$f(x;x_0)=\frac{f(x)-f(x_0)}{x-x_0}$$

и выразим из нее $f(x)$:

$$f(x)=f(x_0)+\left(x-x_0\right)f\left(x;x_0\right). \qquad (1.5.1)$$

Легко заметить, что первое слагаемое в правой части — это интерполяционный полином нулевой степени, а второе слагаемое — это погрешность полинома.

Теперь запишем вторую разделенную разность

$$f\left(x;x_0;x_1\right)=\frac{f\left(x;x_0\right)-f\left(x_0;x_1\right)}{x-x_1},$$

выразим из нее первую разность через вторую и подставим в формулу (1.5.1)

$$f(x)=f\left(x_0\right)+\left(x-x_0\right)f\left(x_1;x_0\right)+\left(x-x_0\right)\left(x-x_1\right)f\left(x;x_0;x_1\right). \qquad (1.5.2)$$

Продолжая этот процесс и выражая вторую разность через третью, третью через четвертую и т. д., получаем:

$$\begin{aligned}f(x)&=f\left(x_0\right)+\left(x-x_0\right)f\left(x_1;x_0\right)+\left(x-x_0\right)\left(x-x_1\right)f\left(x_0;x_1;x_2\right)+\ldots+\\&+\left(x-x_0\right)\left(x-x_1\right)\ldots\left(x-x_{m-1}\right)f\left(x_0;x_1;x_2;\ldots;x_m\right)+\\&+\left(x-x_0\right)\left(x-x_1\right)\ldots\left(x-x_m\right)f\left(x;x_0;x_1;x_2;\ldots;x_m\right)=\\&=Q_m(x)+\omega(x)f\left(x;x_0;x_1;x_2;\ldots;x_m\right).\end{aligned} \qquad (1.5.3)$$

Подставляя $x = x_0$ и $x = x_1$ в (1.5.2), убеждаемся в том, что первое слагаемое в правой части — это интерполяционный полином нулевой степени, а сумма первых двух слагаемых — это интерполяционный полином первой степени. Такая структура сохраняется и в (1.5.3). Сумма первых k слагаемых порождает интерполяционный полином степени $k-1$, а последнее слагаемое является погрешностью интерполяционного полинома степени m. При этом структура полинома такова, что полином степени m получается как полином степени $m-1$ с добавлением еще одного слагаемого

$$\begin{aligned} Q_m(x) &= f(x_0) + (x - x_0) f(x_1; x_0) + (x - x_0)(x - x_1) f(x_0; x_1; x_2) + \ldots + \\ &+ (x - x_0)(x - x_1)\ldots(x - x_{m-1}) f(x_0; x_1; x_2; \ldots; x_m) = \\ &= Q_{m-1}(x) + (x - x_0)(x - x_1)\ldots(x - x_{m-1}) f(x_0; x_1; x_2; \ldots; x_m). \end{aligned} \qquad (1.5.4)$$

Этот полином в форме (1.5.4) и получил название *интерполяционный полином Ньютона*. На практике вычисление разделенных разностей производится в рамках следующей таблицы, где появление новой разделенной разности более высокого порядка связано с построением еще одной диагонали *(см. разд. П1.2)*.

x_0	$f(x_0)$	$f(x_0; x_1)$	$f(x_0; x_1; x_2)$	$f(x_0; x_1; x_2; x_3)$	$f(x_0; x_1; x_2; x_3; x_4)$
x_1	$f(x_1)$	$f(x_1; x_2)$	$f(x_1; x_2; x_3)$	$f(x_1; x_2; x_3; x_4)$	
x_2	$f(x_2)$	$f(x_2; x_3)$	$f(x_2; x_3; x_4)$		
x_3	$f(x_3)$	$f(x_3; x_4)$			
x_4	$f(x_4)$				

Следующее весьма простое замечание дополняет вопрос 3.

Вопрос 5

По заданной таблице при отсутствии ошибок округления построили два полинома: полином Лагранжа (1.3.3) и полином Ньютона (1.5.4). Какой из этих полиномов лучше приближает исходную функцию?

Если относительно функции $f(x)$ сделать те же предположения, что и в формулировке теоремы об остаточном члене интерполяционного полинома в форме Лагранжа, то на основе (1.3.5) и (1.5.3) можно записать

$$\omega(x)f\left(x;x_0;x_1;x_2;...;x_m\right)=\frac{f^{(m+1)}(\eta)}{(m+1)!}\omega(x)$$

или

$$f\left(x;x_0;x_1;x_2;...;x_m\right)=\frac{f^{(m+1)}(\eta)}{(m+1)!}$$

и установить связь разделенной разности с производной. При $m=0$ эта формула превращается в результат теоремы Лагранжа о конечном приращении

$$f(x;x_0)=\frac{f(x)-f(x_0)}{x-x_0}=f'(\eta).$$

Полином Ньютона можно переписать в другой форме, если перенумеровать узлы интерполяции:

- исходная нумерация: $x_0, x_1, ..., x_{n-1}, x_n$;
- новая нумерация: $x_n, x_{n-1}, ..., x_1, x_0$.

В исходной нумерации полином (1.5.4) часто называют полиномом Ньютона для интерполирования вперед или в начале таблицы, а в новой нумерации — полиномом для интерполирования назад или в конце таблицы.

При построении полинома Ньютона узлы таблицы могли задаваться произвольно. Важным и распространенным частным случаем является равномерное расположение узлов. Здесь вместо разделенных разностей можно использовать конечные разности и тем самым избежать деления разности значений функции на разность значений аргумента.

Пусть $x_k=x_0+kh$, где h — постоянный шаг таблицы. Выполним замену переменной $x=x_0+ht$ в (1.5.4), одновременно выражая разделенные разности через конечные по формуле (П1.7). Учитывая, что

$$\frac{x-x_k}{h}=\frac{x-x_0-kh}{h}=\frac{x-x_0}{h}-k=t-k,$$

получим версию полинома Ньютона для равноотстоящих узлов

$$Q_m\left(x_0+ht\right)=f(x_0)+\frac{t}{1!}\Delta f(x_0)+\frac{t(t-1)}{2!}\Delta^2 f(x_0)+...+ \\ +\frac{t(t-1)...(t-m+1)}{m!}\Delta^m f(x_0). \qquad (1.5.5)$$

Новая независимая переменная t принимает в узлах таблицы целые значения, а вычисление конечных разностей реализуется подобно разделенным разностям по следующей таблице:

x_0	f_0	Δf_0	$\Delta^2 f_0$	$\Delta^3 f_0$	$\Delta^4 f_0$
x_1	f_1	Δf_1	$\Delta^2 f_1$	$\Delta^3 f_1$	
x_2	f_2	Δf_2	$\Delta^2 f_2$		
x_3	f_3	Δf_3			
x_4	f_4				

Очевидно, что и полином Ньютона для интерполирования назад также можно записать в аналогичном виде, только замена переменной будет другой $x = x_m + hq$. Тогда

$$\frac{x - x_k}{h} = \frac{x - x_m + kh}{h} = q + k,$$

$$Q_m^-(x) = f(x_m) + \frac{q}{1!}\Delta f_{m-1} + \frac{q(q+1)}{2!}\Delta^2 f_{m-1} + \ldots + \frac{q(q+1)\ldots(q+m-1)}{m!}\Delta^m f_0,$$

$$R_m^-(x) = h^{m+1}\frac{q(q+1)\ldots(q+m)}{(m+1)!}f^{(m+1)}(\eta).$$

Рассмотренные интерполяционные полиномы не исчерпывают всего их многообразия. Для различных целей могут быть использованы формулы Гаусса, Стирлинга, Бесселя и др., ознакомиться с которыми можно по приведенной литературе.

1.6. Интерполирование сплайнами

На практике интерполяционные полиномы высоких степеней строят крайне редко. В первую очередь, это связано с тем, что их коэффициенты очень чувствительны к погрешностям исходных данных. Сравнительно малое изменение узлов интерполирования x_k или значений функции $f(x_k)$ приводит к сильному изменению вида самого полинома. В такой ситуации одним из возможных вариантов аппроксимации является разбиение большой исходной

таблицы на участки, для каждого из которых строится интерполяционный полином относительно невысокой степени. Этот подход используется, например, при получении составных квадратурных формул, рассматриваемых далее. Однако в целом ряде приложений требуется, чтобы аппроксимирующая функция была гладкой, а функция, составленная из различных полиномов, в узлах сопряжения не имела производной. Выходом из создавшегося положения является использование сплайн-интерполяции.

Сплайн (от англ. *spline*) — это длинная гибкая тонкая рейка, используемая чертежниками в качестве лекала для проведения гладких кривых через заданные точки. Математическое осмысление этого инструмента и породило теорию сплайнов, аппарат которой заметно выходит за рамки описания механического сплайна. Расположив чертеж в вертикальной плоскости и закрепив рейку, в узлах интерполяции к ней подвешивают грузила и добиваются, чтобы деформированная рейка совместилась со всеми точками. Сплайн принимает форму, отвечающую минимуму его потенциальной энергии, и линеаризованное дифференциальное уравнение изогнутой оси рейки имеет вид

$$EI \cdot s''(x) = -M(x),$$

где EI — жесткость материала рейки, $s(x)$ — ее прогиб, а $M(x)$ — изгибающий момент, линейно зависящий от координаты ($M(x) \sim x$). Интегрирование этого уравнения показывает, что функция $s(x)$, описывающая профиль сплайна, является кубическим полиномом между любыми двумя соседними точками. Кроме этого, соседние полиномы соединяются непрерывно и гладко так же, как и их первые и вторые производные (рейка не разламывается).

От механической иллюстрации перейдем к формальному аппарату сплайн-интерполяции. Обратимся к таблично заданной функции:

x	x_1	x_2	x_3	...	x_N
$f(x)$	$f(x_1)$	$f(x_2)$	$f(x_3)$	...	$f(x_N)$

Число узлов равно N, а их нумерация начинается с единицы. На каждом промежутке $[x_k, x_{k+1}]$ будем строить интерполяционный полином третьей степени

$$S_k(x) = a_k + b_k(x - x_k) + c_k(x - x_k)^2 + d_k(x - x_k)^3. \qquad (1.6.1)$$

Количество полиномов, как и промежутков, равно $N-1$, и каждый полином имеет 4 параметра. Таким образом, всего в наличии $4N-4$ параметра. Потребуем, чтобы во всех внутренних точках были равны значения соседних полиномов, их первых и вторых производных

$$S_k(x_{k+1}) = S_{k+1}(x_{k+1});\ S'_k(x_{k+1}) = S'_{k+1}(x_{k+1});$$
$$S''_k(x_{k+1}) = S''_{k+1}(x_{k+1});\ k = 1, ..., N-2, \tag{1.6.2}$$

т. е. выполнялись суммарно $3(N-2) = 3N-6$ уравнений. Еще N уравнений отражают требования интерполирования

$$S_k(x_k) = f_k, \quad k = 1, 2, ..., N-1; \quad S_{N-1}(x_N) = f_N. \tag{1.6.3}$$

Общее число задаваемых условий достигает $4N-6$. При наличии $4N-4$ параметров появляется возможность выполнить еще два условия. В их задании нет острой необходимости, т. к. требования интерполирования и сопряжения соседних полиномов уже выполнены, но это целесообразно сделать для однозначного решения задачи. Различные кубические сплайны отличаются друг от друга заданием этих двух требований, которые, как правило, записываются для двух крайних точек x_1 и x_N. К этим двум дополнительным условиям целесообразно предъявить следующие два требования. С одной стороны, их лучше задавать так, чтобы полная система уравнений решалась по возможности более просто. С другой стороны, они должны максимально соответствовать характеру поведения функции в начале и в конце промежутка интерполирования. Приведем два примера.

Пример 1. $S''_1(x_1) = 0$; $S''_{N-1}(x_N) = 0$. Этот сплайн получил название *естественного кубического сплайна*. Однако такое название оправдывается, и выполнение второго требования имеет место только в механике, пока анализируется упомянутый механический сплайн. В общем случае равенство нулю второй производной на краях промежутка не является обязательным свойством экспериментальных данных, отражаемых таблицей.

Пример 2. По первым четырем точкам таблицы строится интерполяционный полином третьей степени $Q_3(x)$, и его третья производная приравнивается третьей производной $S_1(x)$. Аналогично по последним четырем точкам строится интерполяционный полином $\tilde{Q}_3(x)$, и его третья производная приравнивается третьей производной последнего полинома $S_{N-1}(x)$

$$Q'''_3(x_1) = S'''_1(x_1);\ \tilde{Q}'''_3(x_1) = S'''_{N-1}(x_N). \tag{1.6.4}$$

Такие условия не только отвечают характеру поведения функции в начале и в конце промежутка интерполирования, но и достаточно просты (третья производная от полинома третьей степени постоянна). Именно они и учитываются в рассматриваемых программах SPLINE и SEVAL. Процедура решения системы линейных алгебраических уравнений (1.6.2)—(1.6.4) для $4N-4$ неизвестных коэффициентов полиномов (1.6.1) может быть заметно упрощена. Специальная форма записи (1.6.1) уже позволяет сократить число неизвестных на четверть, т. к. из требования интерполирования сразу находятся неизвестные коэффициенты a_k

$$S_k(x_k) = a_k = f_k, \qquad k = 1, 2, ..., N-1. \tag{1.6.5}$$

Для дальнейшего уменьшения объема вычислений запишем полиномы (1.6.1) на промежутке $[x_k, x_{k+1}]$ в несколько ином виде. С этой целью введем обозначения

$$h_k = x_{k+1} - x_k, \quad w = \frac{x - x_k}{h_k}, \quad \overline{w} = 1 - w = \frac{x_{k+1} - x}{h_k}. \tag{1.6.6}$$

Переменная w изменяется на $[x_k, x_{k+1}]$ от 0 до 1, а $\overline{w}$ — от 1 до 0. В этих обозначениях запишем $S_k(x)$ в виде

$$S_k(x) = w \cdot f_{k+1} + \overline{w} \cdot f_k + h_k^2\left[\left(w^3 - w\right)\sigma_{k+1} + \left(\overline{w}^3 - \overline{w}\right)\sigma_k\right], \tag{1.6.7}$$

где σ_k и σ_{k+1} — константы, подлежащие определению, $k = 1, ..., N-1$.

Первые два слагаемых являются интерполяционным полиномом первой степени и обеспечивают тем самым требования интерполирования ($S_k(x_k) = f_k$, $S_k(x_{k+1}) = f_{k+1}$), а также сопряжение соседних полиномов. Остальные слагаемые (1.6.7) в узлах интерполирования равны нулю. Их цель — обеспечить сопряжение первых и вторых производных соседних полиномов.

Определим первые три производные $S_k(x)$, учитывая, что $w' = 1/h_k$, а $\overline{w}' = -1/h_k$:

$$S_k'(x) = \left(f_{k+1} - f_k\right)/h_k + h_k\left[(3w^2 - 1)\sigma_{k+1} - (3\overline{w}^2 - 1)\sigma_k\right],$$

$$S_k''(x) = 6w\sigma_{k+1} + 6\overline{w}\sigma_k; \; S_k'''(x) = 6\frac{(\sigma_{k+1} - \sigma_k)}{h_k}. \tag{1.6.8}$$

Непрерывность вторых производных ($S_k''(x_{k+1}) = S_{k+1}''(x_{k+1})$) очевидна после подстановки в (1.6.8) $x = x_k$ и $x = x_{k+1}$. Осталось обеспечить сопряжение по первым производным.

$$S_k'(x_{k+1}) = \left(f_{k+1} - f_k\right)/h_k + h_k\left(2\sigma_{k+1} + \sigma_k\right) = f(x_k; x_{k+1}) + h_k\left(2\sigma_{k+1} + \sigma_k\right),$$

$$S_{k+1}'(x_{k+1}) = \left(f_{k+2} - f_{k+1}\right)/h_{k+1} + h_{k+1}\left(-\sigma_{k+2} - 2\sigma_{k+1}\right) =$$
$$= f(x_{k+1}; x_{k+2}) - h_{k+1}\left(\sigma_{k+2} + 2\sigma_{k+1}\right).$$

Требование этого сопряжения ($S_k'(x_{k+1}) = S_{k+1}'(x_{k+1})$) приводит к следующей системе уравнений:

$$\begin{aligned} &h_k\sigma_k + 2\left(h_k + h_{k+1}\right)\sigma_{k+1} + h_{k+1}\sigma_{k+2} = f(x_{k+1}; x_{k+2}) - f(x_k; x_{k+1}), \\ &k = 1, 2, \ldots, N-2. \end{aligned} \qquad (1.6.9)$$

Осталось записать в этих же обозначениях уравнения (1.6.4) на краях промежутка интерполирования. В качестве $Q_3(x)$ воспользуемся полиномом Ньютона (1.5.4) третьей степени по первым четырем точкам, начиная с x_1:

$$Q_3(x) = f(x_1) + \left(x - x_1\right) f(x_1; x_2) + \left(x - x_1\right)\left(x - x_2\right) f(x_1; x_2; x_3) + \ldots +$$
$$+ \left(x - x_1\right)\left(x - x_2\right)\left(x - x_3\right) f(x_1; x_2; x_3; x_4).$$

Дифференцируя его три раза ($Q_3'''(x_1) = 6f(x_1; x_2; x_3; x_4)$) и учитывая (1.6.8), получаем для первого уравнения (1.6.4)

$$-h_1\sigma_1 + h_1\sigma_2 = h_1^2 f(x_1; x_2; x_3; x_4). \qquad (1.6.10)$$

Аналогично для второго уравнения (1.6.4), трижды дифференцируя полином Ньютона третьей степени по последним четырем точкам таблицы, имеем

$$\begin{aligned} &-h_1\sigma_1 + h_1\sigma_2 = h_1^2 f(x_1; x_2; x_3; x_4) \\ &h_{N-1}\sigma_{N-1} - h_{N-1}\sigma_N = -h_{N-1}^2 f(x_{N-3}; x_{N-2}; x_{N-1}; x_N). \end{aligned} \qquad (1.6.11)$$

Уравнения (1.6.10) и (1.6.11) в совокупности с системой (1.6.9) образуют систему из N алгебраических уравнений с N неизвестными σ_k. При этом (1.6.10) является первым уравнением системы, а (1.6.11) — последним. Матрица такой системы является симметрической, трехдиагональной и диаго-

нально доминирующей (в каждой строке диагональный элемент больше суммы остальных элементов):

$$\begin{pmatrix} -h_1 & h_1 & 0 & 0 & \dots & 0 \\ h_1 & 2(h_1+h_2) & h_2 & 0 & \dots & 0 \\ 0 & h_2 & 2(h_2+h_3) & h_3 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & \dots & 0 & h_{N-2} & 2(h_{N-2}+h_{N-1}) & h_{N-1} \\ 0 & \dots & 0 & 0 & h_{N-1} & -h_{N-1} \end{pmatrix}.$$

Для решения линейных систем с трехдиагональной матрицей существует эффективный численный алгоритм, называемый *методом прогонки* и рассматриваемый далее в соответствующем разделе. Трудоемкость обычного метода Гаусса пропорциональна кубу размера матрицы, а для метода прогонки она лишь линейно зависит от числа уравнений системы.

В силу целого ряда соображений, результаты решения системы уравнений относительно σ_k часто хранят в виде коэффициентов b_k, c_k, d_k полинома (1.6.1). С учетом (1.6.8) эти коэффициенты легко записываются через σ_k:

$$b_k = S_k'(x_k) = \left(f_{k+1} - f_k\right)/h_k - h_k\left(\sigma_{k+1} + 2\sigma_k\right);$$

$$c_k = \frac{1}{2}S_k''(x_k) = 3\sigma_k;\ d_k = \frac{1}{6}S_k'''(x) = \frac{\sigma_{k+1} - \sigma_k}{h_k}.$$

Программное обеспечение, с текстами которого, как и с другими программами, обсуждаемыми в данной книге, можно ознакомиться в работе [14], состоит из двух процедур. Первая из них:

```
SPLINE(N, X, F, B, C, D)
```

оформленная как процедура, решает систему уравнений относительно σ_k.

Здесь:

- `N` — число точек;
- `X` и `F` — векторы, элементами которых являются x_k и f_k;
- `B`, `C`, `D` — векторы с коэффициентами b_k, c_k, d_k полиномов (1.6.1) — результаты работы `SPLINE`.

Вторая программа:

```
SEVAL(N, U, X, F, B, C, D)
```

оформленная как функция, использует результаты работы SPLINE и вычисляет значение сплайна в заданной точке U.

1.7. Интерполяционный полином Эрмита

Все предыдущие разделы этой главы относились к задаче интерполирования по значениям функции. Если в таблице помимо значений функции присутствуют ее производные и от интерполяционного полинома требуется совпадение с данными этой таблицы, то такая задача называется *интерполированием по Эрмиту*. Познакомимся с ней на отдельных примерах.

Для следующих исходных данных:

x	x_0	x_1	x_2
$f(x)$	$f(x_0)$	$f(x_1)$	$f(x_2)$
$f'(x)$	$f'(x_0)$	$f'(x_1)$	
$f''(x)$		$f''(x_1)$	

требуется построить интерполяционный полином $H(x)$, удовлетворяющий всем условиям таблицы. Прежде чем это будет сделано, целесообразно ответить на некоторые вопросы.

Вопрос 6

Какой степени получится полином Эрмита? Как найти его коэффициенты? Если по этим же трем узлам построить полином Лагранжа, используя только значения f_k, то какой из двух полиномов будет приближать функцию успешнее и почему?

Потребуем от $H(x)$ выполнения условий таблицы

$$\begin{aligned} &H(x_k)=f(x_k), \quad k=0,1,2; \\ &H'(x_k)=f'(x_k), \quad k=0,1; \\ &H''(x_k)=f''(x_k), \quad k=1. \end{aligned} \tag{1.7.1}$$

Система (1.7.1) содержит шесть уравнений. Для ее однозначного решения полином $H(x)$ должен иметь 6 коэффициентов, т. е. быть полиномом пятой степени. Общее правило очевидно: *степень интерполяционного полинома Эрмита на единицу меньше общего числа условий таблицы.*

В задаче интерполирования по значениям функции уже возникала система (1.2.2), аналогичная системе (1.7.1). Тогда оказалось возможным построить, например, полином Лагранжа (1.3.3), не решая системы (1.2.2). Нельзя ли и в случае полинома Эрмита воспроизвести его сразу в готовом виде, не решая системы, подобной (1.7.1)? Ответ на этот вопрос положительный. Однако общая формула является громоздкой, и мы не будем ее приводить. При этом форма записи полинома будет проще, если исходная таблица симметричная, т. е. число и вид условий во всех узлах одинаковые (в отличие от системы (1.7.1)). Так, например, если в каждом узле x_0, x_1, ..., x_m заданы функция $f(x_k)$ и ее производная $f'(x_k)$, то полином Эрмита имеет степень $2m+1$ и описывается следующей формулой:

$$\begin{aligned} H_{2m+1}(x)=\sum_{k=0}^{m}&\left[\frac{\omega(x)}{(x-x_k)\omega'(x_k)}\right]^2\times \\ &\times\left[f(x_k)\cdot\left(1-\frac{\omega''(x_k)}{\omega'(x_k)}(x-x_k)\right)+f'(x_k)(x-x_k)\right], \end{aligned} \tag{1.7.2}$$

а его остаточный член отвечает выражению:

$$R^{Э}_{2m+1}(x)=\frac{f^{(2m+2)}(\eta)}{(2m+2)!}\omega^2(x), \qquad \eta\in[a,b]. \tag{1.7.3}$$

В обеих формулах $\omega(x)$ по-прежнему имеет вид (1.3.1).

В качестве еще одной иллюстрации обратимся к разложению функции $f(x)$ в степенной ряд Тейлора в точке x_0

$$\begin{aligned} f(x) &= f(x_0) + \frac{x - x_0}{1!} f'(x_0) + \\ &+ \frac{(x - x_0)^2}{2!} f''(x_0) + \frac{(x - x_0)^3}{3!} f'''(\eta) = \\ &= H_2(x) + \frac{(x - x_0)^3}{3!} f'''(\eta). \end{aligned} \tag{1.7.4}$$

Полином второй степени $H_2(x)$, с одной стороны, является частной суммой ряда Тейлора, а с другой стороны, удовлетворяет условиям

$$H_2(x_0) = f(x_0),\ H_2'(x_0) = f'(x_0),\ H_2''(x_0) = f''(x_0),$$

что позволяет назвать его одновременно и интерполяционным полиномом Эрмита с одним узлом интерполирования.

В заключение данного раздела зададим еще один вопрос, дополняющий вопросы 3 и 5. Надеемся, что на него читатели смогут ответить самостоятельно.

Вопрос 7

На промежутке $[x_0, x_2]$ по трем узлам интерполирования x_0, x_1, x_2 построим интерполяционный полином Лагранжа второй степени $Q_2(x)$, а затем построим полином Эрмита $H_2(x)$, в соответствии с формулой (1.7.4). Оба полинома являются полиномами второй степени. Какой из них приближает исходную функцию $f(x)$ лучше и почему?

1.8. Обратная интерполяция

До данного раздела решалась прямая задача интерполирования. По известной таблице:

x	x_1	x_1	x_2	...	x_m
$f(x)$	$f(x_0)$	$f(x_1)$	$f(x_2)$	...	$f(x_m)$

и заданному значению x^* требуется оценить значение функции $f(x^*)$. В обратной задаче для такой же таблицы требуется восстановить значение аргумента x^*, для которого функция принимает заданное значение f^*. На практике чаще всего используется один из следующих очевидных способов.

Способ 1. Меняются местами строки таблицы, в качестве узлов интерполирования выбираются значения f_k, а в качестве значений функции x_k, и строится интерполяционный полином для обратной функции. Подставляя в него f^*, находим желаемое x^*. Такой подход вполне жизнеспособен, если существует обратная функция, т. е. исходная функция является строго монотонной. В противном случае необходимо делить таблицу на части с участками строгой монотонности.

Например, обратный квадратичный полином для фрагмента таблицы:

$f(x)$	$f(x_{k-2})$	$f(x_{k-1})$	$f(x_k)$
x	x_{k-2}	x_{k-1}	x_k

будет иметь вид:

$$L_2(f)=\frac{(f-f_{k-1})(f-f_k)}{(f_{k-2}-f_{k-1})(f_{k-2}-f_k)}x_{k-2}+$$
$$+\frac{(f-f_{k-2})(f-f_k)}{(f_{k-1}-f_{k-2})(f_{k-1}-f_k)}x_{k-1}+\frac{(f-f_{k-2})(f-f_{k-1})}{(f_k-f_{k-2})(f_k-f_{k-1})}x_k.$$

Способ 2. По исходной таблице строится обычный интерполяционный полином $Q_m(x)$ с узлами x_k, а затем решается уравнение $Q_m(x)=f^*$. Для полиномов до четвертой степени ответ может быть получен даже аналитически, а в других случаях это уравнение решается численно, например, одним из методов решения нелинейных уравнений, рассматриваемых далее. Здесь нет необходимости принимать во внимание монотонность. Для немонотонной функции уравнение будет иметь несколько корней на промежутке интерполирования, и из них лишь нужно выбрать отвечающий поставленной задаче.

2 Зак. 3418

1.9. Простейшие квадратурные формулы

Квадратурные формулы — это формулы для вычисления значения определенного интеграла. Их получение — одно из многочисленных возможных приложений интерполяционных полиномов. Пусть требуется вычислить следующий интеграл

$$I = \int_a^b f(x)dx, \tag{1.9.1}$$

точное определение которого весьма затруднено или невозможно. Тогда исходная функция может быть аппроксимирована интерполяционным полиномом:

$$f(x) = Q_m(x) + R_m(x),$$

и интеграл от интерполяционного полинома порождает некоторую квадратурную формулу

$$I = \int_a^b f(x)dx \approx \int_a^b Q_m(x)dx, \tag{1.9.2}$$

а интеграл от остаточного члена полинома определяет ее погрешность

$$\varepsilon = \int_a^b R_m(x)dx. \tag{1.9.3}$$

Будем последовательно подставлять в (1.9.2) полиномы различных степеней, начиная с нулевой ($Q_0(x) = f(x_0)$).

$$I \approx (b-a)f(x_0).$$

Наиболее популярными являются следующие три варианта выбора узла x_0:

$$x_0 = a, \qquad I \approx (b-a)f(a), \tag{1.9.4}$$

$$x_0 = b, \qquad I \approx (b-a)f(b), \tag{1.9.5}$$

$$x_0 = \frac{a+b}{2}, \qquad I \approx (b-a)f\left(\frac{a+b}{2}\right). \tag{1.9.6}$$

Формула (1.9.4) называется квадратурной формулой *левых прямоугольников*, (1.9.5) — формулой *правых прямоугольников*, (1.9.6) — формулой *средних прямоугольников*. Причины таких названий легко понять из геометрических

иллюстраций рис. 1.4, *а*—*в*, откуда видно, что площадь под заданной функцией аппроксимируется площадью соответствующего прямоугольника.

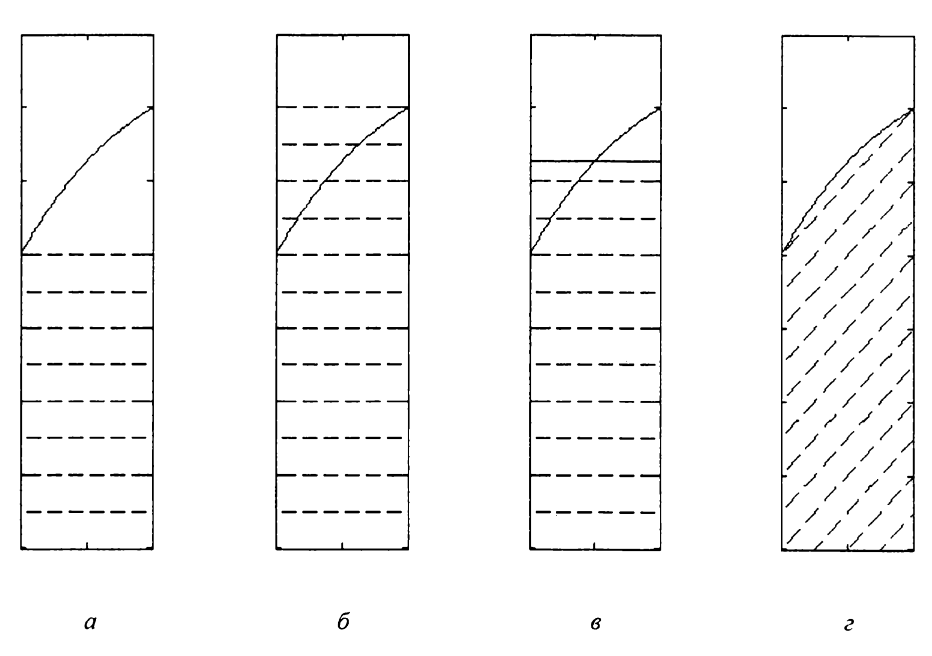


Рис. 1.4. Квадратурные формулы (1.9.4) — (1.9.7)

Интегрирование полинома первой степени с узлами интерполирования $x_0 = a$ и $x_1 = b$

$$Q_1(x) = \frac{x-b}{a-b} f(a) + \frac{x-a}{b-a} f(b)$$

порождает квадратурную *формулу трапеций* (рис. 1.4, *г*)

$$I \approx \frac{b-a}{2}\bigl(f(a) + f(b)\bigr), \qquad (1.9.7)$$

а интегрирование полинома второй степени с узлами $x_0 = a$, $x_1 = \dfrac{a+b}{2}$ и $x_2 = b$ приводит к квадратурной *формуле Симпсона*:

$$I \approx \frac{b-a}{6}\left(f(a) + 4f\left(\frac{a+b}{2}\right) + f(b)\right). \qquad (1.9.8)$$

Аналогичные формулы могут быть получены и для полиномов более высоких степеней. Прежде чем будут получены соответствующие выражения, проверьте свою интуицию и попытайтесь ответить на вопрос, руководствуясь формулами (1.9.6), (1.9.7) и соответствующими рисунками.

Вопрос 8

Многократно для самых различных функций считаются интегралы по формуле средних прямоугольников и по формуле трапеций. Для какой из них "в среднем" значение интеграла будет получаться точнее?

Оценку погрешности квадратурных формул будем выполнять на основе двух известных в математике теорем о среднем.

Теорема. Пусть $f(x)$ и $g(x)$ непрерывны на $[a,b]$, а $g(x)$ в дополнение к этому знакопостоянна. Тогда найдется точка $c \in [a,b]$ такая, что

$$\int_a^b g(x)f(x)dx = f(c)\int_a^b g(x)dx. \qquad (1.9.9)$$

Теорема. Пусть $f(x)$ непрерывна на $[a,b]$, и заданы N точек $x_k \in [a,b]$. Найдется точка $\eta \in [a,b]$ такая, что

$$\frac{1}{N}\sum_{k=1}^{N} f(x_k) = f(\eta). \qquad (1.9.10)$$

Остаточный член интерполяционного полинома нулевой степени имеет вид

$$R_0(x) = \frac{x - x_0}{1!} f'(\eta).$$

Полезно заметить, что точка η зависит от x, т. е. $\eta = \eta(x)$, как это уже отмечалось в конце *разд. 1.3*. Последовательно подставляя $x_0 = a$, b и $\frac{a+b}{2}$, вычислим интеграл (1.9.3).

Формула левых прямоугольников:

$$\varepsilon_{\text{лев. пр.}} = \int_a^b (x-a)f'(\eta)dx = f'(\eta^*)\int_a^b (x-a)dx = \frac{(b-a)^2}{2} f'(\eta^*). \qquad (1.9.11)$$

Формула правых прямоугольников:

$$\varepsilon_{\text{прав. пр.}} = \int_a^b (x-b)f'(\eta)dx = f'(\eta^*)\int_a^b (x-b)dx = -\frac{(b-a)^2}{2}f'(\eta^*). \quad (1.9.12)$$

Формула средних прямоугольников:

$$\varepsilon_{\text{средн. пр.}} = \int_a^b \left(x - \frac{a+b}{2}\right) f'(\eta)dx.$$

В выражениях (1.9.11) и (1.9.12) была использована теорема о среднем (1.9.9), т. к. функции $x-a$ и $x-b$ являются знакопостоянными на $[a,\ b]$. Воспользоваться этой теоремой для средних прямоугольников не удается, т. к. $\left(x - \frac{a+b}{2}\right)$ меняет знак. Для оценки погрешности в этом случае воспользуемся разложением $f(x)$ в ряд в точке $\frac{a+b}{2}$:

$$f(x) = f\left(\frac{a+b}{2}\right) + \frac{x - \frac{a+b}{2}}{1!} f'\left(\frac{a+b}{2}\right) + \frac{\left(x - \frac{a+b}{2}\right)^2}{2!} f'(\eta),$$

интегрируя его на $[a,\ b]$. Интеграл от первого слагаемого дает формулу средних прямоугольников, интеграл от второго слагаемого равен нулю, а интеграл от последнего слагаемого дает оценку погрешности. Таким образом, для формулы средних прямоугольников имеем:

$$\begin{aligned}\varepsilon_{\text{средн. пр.}} &= \frac{1}{2}\int_a^b \left(x - \frac{a+b}{2}\right)^2 f''(\eta)dx = \frac{f''(\eta^*)}{2}\int_a^b \left(x - \frac{a+b}{2}\right)^2 dx = \\ &= \frac{(b-a)^3}{24} f''(\eta^*).\end{aligned} \quad (1.9.13)$$

Использование теоремы о среднем в последнем случае уже вполне закономерно. Формулы (1.9.11) и (1.9.12) отличаются знаком. Возникает следующий вопрос.

Вопрос 9

Дважды вычисляем интеграл по формулам сначала левых, а затем правых прямоугольников. Определяем полусумму результатов. Погрешности сокращаются, и результат получается точным?

Для оценки погрешности формулы трапеций проинтегрируем остаточный член полинома первой степени

$$\varepsilon_{\text{трап}} = \int_a^b \frac{(x-a)(x-b)}{2!} f''(\eta)dx = f''(\eta^*)\int_a^b \frac{(x-a)(x-b)}{2!}dx = \\ = -\frac{(b-a)^3}{12} f''(\eta^*). \qquad (1.9.14)$$

Здесь также применена теорема о среднем. Наконец, для интеграла от остаточного члена интерполяционного полинома второй степени

$$R_2(x) = \frac{(x-a)\left(x-\frac{a+b}{2}\right)(x-b)}{3!} f'''(\eta)$$

условия теоремы о среднем не выполняются, и погрешность формулы Симпсона определяется по-другому. По этим же трем узлам строится полином Эрмита уже третьей степени с двумя условиями в центральной точке, погрешность которого и интегрируется. Результат приведем без вывода.

$$\varepsilon_{\text{Симпс}} = -\frac{(b-a)^5}{2880} f^{(4)}(\eta). \qquad (1.9.15)$$

В большинстве случаев подынтегральная функция не описывается удовлетворительно полиномами первой или второй степени. Поэтому для достижения необходимой точности исходный промежуток разбивается на такие малые промежутки, где указанная аппроксимация удачна, на каждом из этих промежутков применяется выбранная квадратурная формула, а результаты складываются. Такие формулы получили название *составных* (или больших) *квадратурных формул*.

1.9.1. Составные квадратурные формулы

Разобьем исходный промежуток на N равных промежутков $[x_k, x_{k+1}]$

$$h = \frac{b-a}{N}, \quad x_k = x_0 + kh, \quad x_0 = a, \quad x_N = b.$$

На каждом таком промежутке применим формулу левых прямоугольников и результаты сложим:

$$I_k = \int_{x_k}^{x_{k+1}} f(x)dx \approx (x_{k+1} - x_k) f(x_k) = \frac{b-a}{N} f(x_k);$$

$$I_{\text{лев. пр.}} = \sum_{k=0}^{N-1} I_k \approx \frac{b-a}{N} \sum_{k=0}^{N-1} f(x_k). \quad (1.9.16)$$

Аналогичные преобразования для формул правых и средних прямоугольников, а также для формулы трапеций приводят к следующим результатам:

- *правые прямоугольники*:

$$I_k = \int_{x_k}^{x_{k+1}} f(x)dx \approx (x_{k+1} - x_k) f(x_{k+1}) = \frac{b-a}{N} f(x_{k+1});$$

$$I_{\text{прав. пр.}} = \sum_{k=0}^{N-1} I_k \approx \frac{b-a}{N} \sum_{k=0}^{N-1} f(x_{k+1}) = \frac{b-a}{N} \sum_{k=1}^{N} f(x_k); \quad (1.9.17)$$

- *средние прямоугольники*:

$$I_k = \int_{x_k}^{x_{k+1}} f(x)dx \approx (x_{k+1} - x_k) f(x_k + h/2) = \frac{b-a}{N} f(x_k + h/2);$$

$$I_{\text{средн. пр.}} = \sum_{k=0}^{N-1} I_k \approx \frac{b-a}{N} \sum_{k=0}^{N-1} f(x_k + h/2) = \\ = \frac{b-a}{N} \sum_{k=0}^{N-1} f\left(x_k + \frac{b-a}{2N} \right); \quad (1.9.18)$$

- *трапеции*:

$$I_k = \int_{x_k}^{x_{k+1}} f(x)dx \approx \frac{x_{k+1} - x_k}{2} \big(f(x_{k+1}) + f(x_k) \big) = \frac{b-a}{2N} \big(f(x_{k+1}) + f(x_k) \big);$$

$$I_{\text{трап}} = \sum_{k=0}^{N-1} I_k \approx \frac{b-a}{2N} \sum_{k=0}^{N-1} \big(f(x_{k+1}) + f(x_k) \big) = \\ = \frac{b-a}{2N} \left(f(a) + 2 \sum_{k=1}^{N-1} f(x_k) + f(b) \right). \quad (1.9.19)$$

Формулы (1.9.16)—(1.9.19) называются *составными квадратурными формулами* левых, правых, средних прямоугольников и трапеций соответственно. Для получения составной формулы Симпсона будем выбирать N всегда четным, а исходный промежуток разобьем на $N/2$ равных промежутков $[x_k, x_{k+2}]$ длиной $2(b-a)/N$. На каждом из них интеграл равен

$$I_k = \int_{x_k}^{x_{k+2}} f(x)dx \approx \frac{x_{k+2}-x_k}{6}\big(f(x_{k+2})+4f(x_{k+1})+f(x_k)\big) = \\ = \frac{b-a}{3N}\big(f(x_{k+2})+4f(x_{k+1})+f(x_k)\big).$$

Суммируя по всем промежуткам, получаем *составную формулу Симпсона*:

$$I_{\text{Симпс}} = \frac{b-a}{3N}\times \\ \times\left[f(a)+4\left(f_1+f_3+f_5+\ldots+f_{N-1}\right)+2\left(f_2+f_4+\ldots+f_{N-2}\right)+f(b)\right]. \quad (1.9.20)$$

1.9.2. Погрешности составных формул

Для оценки погрешности составных формул вычисляем погрешность каждого малого участка и результаты складываем. Так, для формулы левых прямоугольников на основе (1.9.11), применяя дискретный вариант теоремы о среднем (1.9.10) к выражению в квадратных скобках, имеем:

$$\varepsilon_k = \frac{(x_{k+1}-x_k)^2}{2}f'(\eta_k) = \frac{(b-a)^2}{2N^2}f'(\eta_k),$$

$$\varepsilon_{\text{лев. пр.}} = \sum_{k=1}^{N}\varepsilon_k = \frac{(b-a)^2}{2N}\left[\frac{1}{N}\sum_{k=1}^{N}f'(\eta_k)\right] = \frac{(b-a)^2}{2N}f'(\eta). \quad (1.9.21)$$

Повторяя аналогичные вычисления с формулой (1.9.12) для правых прямоугольников получаем:

$$\varepsilon_{\text{прав. пр.}} = -\frac{(b-a)^2}{2N}f'(\eta). \quad (1.9.22)$$

Выражение (1.9.13) позволяет получить составную формулу средних прямоугольников:

$$\varepsilon_k = \frac{\left(x_{k+1} - x_k\right)^3}{24} f''\left(\eta_k\right) = \frac{(b-a)^3}{24N^3} f''\left(\eta_k\right),$$

$$\varepsilon_{\text{средн. пр.}} = \sum_{k=1}^{N} \varepsilon_k = \frac{(b-a)^3}{24N^2}\left[\frac{1}{N}\sum_{k=1}^{N} f''(\eta_k)\right] = \frac{(b-a)^2}{24N^2} f''(\eta), \qquad (1.9.23)$$

а выражение (1.9.14) — аналогичную формулу трапеций:

$$\varepsilon_k = -\frac{\left(x_{k+1} - x_k\right)^3}{12} f''\left(\eta_k\right) = -\frac{(b-a)^3}{12N^3} f''\left(\eta_k\right),$$

$$\varepsilon_{\text{трап}} = \sum_{k=1}^{N} \varepsilon_k = -\frac{(b-a)^3}{12N^2}\left[\frac{1}{N}\sum_{k=1}^{N} f''\left(\eta_k\right)\right] = -\frac{(b-a)^2}{12N^2} f''(\eta). \qquad (1.9.24)$$

Теперь сравнение формул (1.9.23) и (1.9.24) позволяет вернуться к вопросу 8.

При использовании выражения (1.9.15) для оценки погрешности составной формулы Симпсона следует учитывать, что длина каждого малого участка в два раза больше, чем ранее, а число таких участков в два раза меньше — $N/2$.

$$\varepsilon_k = -\frac{\left(x_{k+2} - x_k\right)^5}{2880} f^{(4)}\left(\eta_k\right) = -\frac{(b-a)^5}{90N^5} f^{(4)}\left(\eta_k\right),$$

$$\varepsilon_{\text{трап}} = \sum_{k=1}^{N/2} \varepsilon_k = -\frac{(b-a)^5}{180N^4}\left[\frac{1}{N/2}\sum_{k=1}^{N/2} f^{(4)}\left(\eta_k\right)\right] = -\frac{(b-a)^5}{180N^4} f^{(4)}(\eta). \qquad (1.9.25)$$

При сравнении формул (1.9.21)—(1.9.25) наибольший интерес вызывает зависимость погрешности от N, позволяющая оценить дополнительный объем вычислений, связанный с увеличением N для уменьшения погрешности до требуемой величины. Нетрудно также заметить, что все эти формулы описываются общей зависимостью

$$\varepsilon = \alpha \frac{(b-a)^{p+1}}{N^p} f^{(p)}(\eta), \qquad (1.9.26)$$

где α — некоторое число. Так для формул левых и правых прямоугольников $p=1$, для средних прямоугольников и трапеций $p=2$, а для формулы Симпсона $p=4$.

Следует заметить, что все квадратурные формулы были получены в предположении, что пределы интегрирования конечны, а подынтегральная функция не имеет особенностей. При вычислении значений несобственных интегралов обычно предварительно с помощью специальных приемов интеграл приводится к специальному виду, позволяющему сочетать использование квадратурной формулы и некоторого аналитического приема. При этом успех часто определяется квалификацией пользователя и его умением учесть специфику решаемой задачи.

В заключение данного раздела остановимся на том, как погрешность квадратурных формул оценивается на практике. Целесообразно назвать следующие способы.

Способ 1. Непосредственная оценка погрешности по формуле (1.9.26) применяется крайне редко из-за известных трудностей, связанных с оценкой производной, особенно для табличной функции, а также из-за отсутствия информации о расположении точки η.

Способ 2. Сравнение результатов, полученных по выбранной составной квадратурной формуле для N и $2N$. В случае неудовлетворительного совпадения значение N удваивается до тех пор, пока требуемая точность не будет достигнута.

Способ 3. Если по каким-либо причинам есть ограничение на рост величины N и способ 2 применен быть не может, сравнивают результаты для одного и того же значения N, но для различных квадратурных формул.

Способ 2 наиболее популярен, и построенная на его основе упрощенная версия алгоритма легко укладывается в несколько элементарных шагов:

1. Задаемся начальным значением N и вычисляем интеграл по любой составной квадратурной формуле, присваивая результат переменной I_{old}.
2. Удваиваем величину N и вновь вычисляем интеграл, присваивая результат переменной I_{new}.
3. Если значения I_{old} и I_{new} совпали с заданной точностью, то прекращаем работу. В противном случае присваиваем переменной I_{old} значение I_{new} и возвращаемся к шагу 2.

Несмотря на весьма простой вид, этот алгоритм позволяет достичь требуемой точности при вычислении интеграла от любой "разумной" функции, не имеющей особенностей на промежутке интегрирования. Более того, надежность написанной на его основе программы вполне сравнима с надежностью программного обеспечения, обсуждаемого в *разд. 1.11*.

1.10. Общий подход к построению квадратурных формул. Метод неопределенных коэффициентов

Все полученные ранее простейшие квадратурные формулы (не составные) имеют следующий вид:

$$\int_a^b f(x)dx \approx \sum_{k=1}^{S} A_k f(x_k). \qquad (1.10.1)$$

Узлы x_k и *веса* A_k квадратурной формулы в предыдущем разделе получались на основе интегрирования соответствующих интерполяционных полиномов. Поставим задачу несколько иначе. Требуется выбрать x_k и A_k так, чтобы формула (1.10.1) была *точной* для полиномов заданной степени. Логика таких требований очевидна. Если подынтегральная функция хорошо аппроксимируется этим полиномом (не обязательно интерполяционным), то и формула (1.10.1) обеспечит требуемую погрешность решения задачи. В противном случае промежуток $[a, b]$ всегда можно разбить на достаточно малые промежутки, применить составные квадратурные формулы и добиться желаемой точности.

Потребуем, чтобы формула (1.10.1) была точна для полинома нулевой степени $f(x) = \alpha = \text{const}$. Выносся константу α из-под знаков интеграла и суммы и сокращая на нее, имеем:

$$\sum_{k=1}^{S} A_k = b - a.$$

Второе уравнение получим, требуя точности (1.10.1) для полинома первой степени и подставляя с этой целью $f(x) = x$:

$$\sum_{k=1}^{S} A_k x_k = \frac{b^2 - a^2}{2}.$$

Это же требование для $f(x) = x^2$ выглядит следующим образом:

$$\sum_{k=1}^{S} A_k x_k^2 = \frac{b^3 - a^3}{3},$$

а в общем случае для $f(x) = x^N$ условие с номером $N+1$ приобретает вид:

$$\sum_{k=1}^{S} A_k x_k^N = \frac{b^{N+1} - a^{N+1}}{N+1}.$$

Объединяя все уравнения, получим следующую систему в общем случае нелинейных уравнений:

$$
\begin{aligned}
&1.\quad \sum_{k=1}^{S} A_k = b - a,\\
&2.\quad \sum_{k=1}^{S} A_k x_k = \frac{b^2 - a^2}{2},\\
&3.\quad \sum_{k=1}^{S} A_k x_k^2 = \frac{b^3 - a^3}{3},\\
&\quad\ldots\ldots\ldots\ldots\ldots\ldots,\\
&N+1.\quad \sum_{k=1}^{S} A_k x_k^N = \frac{b^{N+1} - a^{N+1}}{N+1}
\end{aligned}
\qquad (1.10.2)
$$

относительно $2S$ неизвестных x_k и A_k. В связи с нелинейностью возникает целый ряд вопросов, требующих разрешения. *Существует ли решение? Единственно ли оно? Получаются ли* x_k *в этом случае не только вещественными и различными, но и лежащими на* $[a, b]$*?*

Система (1.10.2) является общей для многих семейств квадратурных формул, отличающихся друг от друга дополнительными условиями, накладываемыми на x_k и A_k. Рассмотрим три таких семейства.

1.10.1. Квадратурные формулы Ньютона — Котеса

Узлы квадратурной формулы x_k здесь выбираются равноотстоящими

$$h = \frac{b-a}{S-1}, \qquad x_k = a + (k-1)h, \qquad x_1 = a, \qquad x_S = b.$$

Система (1.10.2) в данном случае является линейной относительно S неизвестных A_k и легко разрешима. Ее определитель является определителем Вандермонда (1.2.3), что обеспечивает единственность решения. Кроме того, равноотстоящие узлы дают некоторые удобства при программировании этих формул. Но не эти моменты являются решающими. Для составных квадратурных формул при удвоении N (числа внутренних промежутков) в половине возникающих узлов x_k значения функций $f(x_k)$ уже вычислялись ранее и могли быть сохранены, что позволяет сократить объем вычислений вдвое.

Как результат, имеем систему (1.10.2) из S уравнений с S неизвестными A_k, и эти квадратурные формулы оказываются гарантированно точными для полиномов степени $N = S - 1$. Нетрудно заметить, что, решая систему последовательно для $S = 1, 2$ и 3, приходим к уже хорошо знакомым формулам прямоугольников, трапеций и Симпсона, получавшимся ранее другим способом.

1.10.2. Квадратурные формулы Чебышева

Для целого ряда приложений, особенно когда значения $f(x_k)$ определены с заметной погрешностью, П. Л. Чебышевым были предложены квадратурные формулы с равными весами ($A_k = A = \text{const}$):

$$\int_a^b f(x)dx \approx A\sum_{k=1}^{S} f(x_k). \qquad (1.10.3)$$

Система (1.10.2) оказывается системой из $S+1$ уравнения относительно такого же числа неизвестных x_k и A, что позволяет в случае успешного решения получить формулы, гарантированно точные для полиномов степени $N = S$. Однако, учитывая нелинейность системы, вопросы существования и единственности решения выходят на первый план. Оказывается, что система (1.10.2) благополучно решается единственным образом для S от 1 до 7 и для $S = 9$. С. Н. Бернштейном было показано, что для других значений S формулы Чебышева не существуют. Для практических целей это не является серьезным препятствием, т. к. необходимая точность обеспечивается использованием составных квадратурных формул.

1.10.3. Квадратурные формулы Гаусса

В этих формулах на узлы и веса не накладываются никакие дополнительные условия, и все свободные $2S$ параметров используются при решении системы (1.10.2) из $2S$ уравнений. В отличие от формул Чебышева формулы Гаусса существуют для любого числа узлов. Они гарантированно точны для полиномов степени $N = 2S - 1$ и называются формулами *наивысшей алгебраической степени точности.* Далее будет показано, что узлы x_k являются нулями так называемых ортогональных полиномов Лежандра.

Пример. В качестве иллюстрации построим формулу Гаусса с двумя узлами (т. е. $S = 2$) на промежутке $[-1,\ 1]$. Система (1.10.2) задает следующие четыре уравнения:

$$A_1 + A_2 = 2, \quad A_1x_1 + A_2x_2 = 0, \quad A_1x_1^2 + A_2x_2^2 = \frac{2}{3}, \quad A_1x_1^3 + A_2x_2^3 = 0.$$

Умножая второе уравнение на x_1^2 и вычитая результат из четвертого, имеем:

$$A_2x_2\left(x_2^2 - x_1^2\right) = 0\,.$$

Легко убедиться в том, что $A_2 \neq 0$, $x_2 \neq 0$, $x_2 \neq x_1$, и в качестве решения на основе первого и третьего уравнений имеем:

$$x_2 = -x_1 = \frac{1}{\sqrt{3}}; \quad A_1 = A_2 = 1\,.$$

Эта формула точна для полинома третьей степени ($N = 2S - 1 = 3$), и, хотя этот факт нами строго доказан, геометрическая иллюстрация достаточно впечатляющая. График полинома третьей степени на промежутке $[-1,\ 1]$ может быть весьма разнообразным. Полином может иметь здесь как один, так и три нуля, или вообще быть знакопостоянным. Как результат, строим график *произвольного* полинома третьей степени, вычисляем два его значения в точках $\pm 1/\sqrt{3}$, складываем результаты, и значение интеграла от полинома (т. е. площадь под ним) на промежутке $[-1,\ 1]$... оказывается абсолютно точным!

В заключение заметим, что на практике для получения A_k и x_k нет никакой необходимости каждый раз обращаться к системе (1.10.2). Результаты ее решения для стандартного промежутка $[-1,\ 1]$ или $[0,\ 1]$ и для каждого конкретного семейства формул приведены в многочисленных справочниках и

учебниках. Пользователь лишь ограничивается заменой переменных в (1.10.1). Так для промежутка $[-1,\ 1]$ такая замена переменных выглядит следующим образом: $x=\frac{a+b}{2}+\frac{b-a}{2}t$. При изменении t от -1 до 1 переменная x пробегает значения от a до b. С учетом того, что $dx=\frac{b-a}{2}dt$, формула (1.10.1) имеет вид:

$$\begin{aligned}\int_a^b f(x)dx &= \frac{b-a}{2}\int_{-1}^{1} f\left(\frac{a+b}{2}+\frac{b-a}{2}t\right)dt \approx \\ &\approx \frac{b-a}{2}\sum_{k=1}^{S} A_k f\left(\frac{a+b}{2}+\frac{b-a}{2}t_k\right),\end{aligned} \tag{1.10.4}$$

где веса A_k и узлы t_k квадратурной формулы берутся из справочника для $[-1,\ 1]$.

Рассмотренные семейства квадратурных формул не исчерпывают всех возможностей формулы (1.10.1). Как уже отмечалось, другие семейства отличаются выбором дополнительных требований к A_k и x_k.

1.11. Адаптивные квадратурные формулы. Программа *QUANC8*

Простой алгоритм вычисления интеграла на основе составных квадратурных формул, приведенный в конце *разд. 1.9*, отличается достаточной надежностью, но его быстродействие может быть заметно повышено. Например, для функции, представленной на рис. 1.5, на участке с быстрым изменением функции требуется относительно малый шаг. В то же время, т. к. в составной формуле шаг постоянен для всего промежутка, то общий объем вычислений будет неоправданно велик.

Целесообразным представляется построение алгоритма, который был бы способен *адаптироваться* к виду функции и выбирать достаточно малый шаг там, где функция меняется быстро и характеризуется большими производными, и относительно большой шаг там, где функция меняется медленно. На этом пути возможны два варианта: минимизировать погрешность при заданном объеме вычислений или минимизировать объем вычислений при за-

данных требованиях к погрешности. В рассматриваемой программе реализован второй подход.

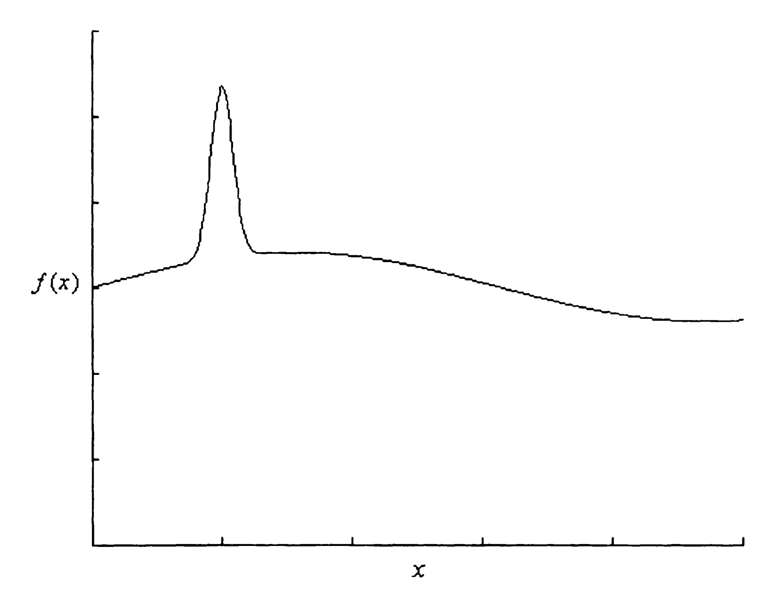


Рис. 1.5. Пример функции с различным поведением

В основу положена квадратурная формула Ньютона — Котеса с девятью узлами, т. е. восемью промежутками между ними, что и оправдывает название программы. Ее составная формула имеет погрешность вида (1.9.26) для $p = 10$. Рассмотрим промежуток длиной h_k внутри $[a, b]$ и введем для него следующие обозначения:

- I_k — точное значение интеграла на этом промежутке;
- P_k — значение интеграла, вычисленное по квадратурной формуле с девятью узлами;
- Q_k — значение интеграла, вычисленное по той же формуле, примененной к двум половинам этого промежутка (по сути используется составная формула с вдвое большим значением N).

Учитывая вид (1.9.26) для $p = 10$, для погрешностей P_k и Q_k последовательно имеем:

$$I_k - P_k \approx 2^p \left(I_k - Q_k\right); \quad I_k \approx \frac{2^p Q_k - P_k}{2^p - 1}; \quad I_k - Q_k \approx \frac{Q_k - P_k}{2^p - 1} = \frac{Q_k - P_k}{1023}.$$

Обозначая требуемую абсолютную погрешность вычисления интеграла на всем промежутке $[a,b]$ за ε_A, считаем промежуток h_k "принятым", а интеграл на нем вычисленным, если выполняется неравенство

$$\left|\frac{Q_k - P_k}{1023}\right| \le \frac{h_k}{b-a}\varepsilon_A. \qquad (1.11.1)$$

Множитель $\frac{h_k}{b-a}$ является весовым коэффициентом и отражает вклад погрешности на промежутке h_k в общую погрешность для всего промежутка. Возможно также использование и относительной погрешности ε_R

$$\left|\frac{Q_k - P_k}{1023}\right| \le \frac{h_k}{b-a}\varepsilon_R \left|\tilde{I}_k\right|, \qquad (1.11.2)$$

где $\tilde{I}_k$ — оценка вычисления интеграла по всему промежутку. Следует, однако, помнить, что использование критерия относительной погрешности усложняется, если значение $\tilde{I}_k$ оказывается нулевым или близким к нулю. В программе QUANC8 пользователю предоставляется возможность выбирать один из трех вариантов контроля погрешности на основе объединения формул (1.11.1) и (1.11.2):

$$\left|\frac{Q_k - P_k}{1023}\right| \le \frac{h_k}{b-a}\max\left(\varepsilon_A\ ;\ \varepsilon_R \left|\tilde{I}_k\right|\right), \qquad (1.11.3)$$

- Вариант 1: $\varepsilon_R = 0$, $\varepsilon_A \ne 0$ — контроль абсолютной погрешности.
- Вариант 2: $\varepsilon_R \ne 0$, $\varepsilon_A = 0$ — контроль относительной погрешности.
- Вариант 3: $\varepsilon_R \ne 0$, $\varepsilon_A \ne 0$ — контроль "смешанной" погрешности.

В последнем случае делается попытка избежать упомянутых неприятных ситуаций со значениями $\tilde{I}_k$, близкими к нулю.

Адаптация программы к виду функции и реализация переменного шага интегрирования реализуются в соответствии со следующим алгоритмом. Вычисляются P_k и Q_k применительно ко всему промежутку. Если погрешность еще достаточно велика, промежуток делится пополам, значения подынтегральной функции $f(x)$, вычисленные на правой половине промежутка, за-

поминаются, и все повторяется для левой половины промежутка. Такое обращение каждый раз к левой половине текущего промежутка продолжается до тех пор, пока крайний слева промежуток не будет принят. После этого обрабатывается ближайший к нему правый промежуток. Запоминание значений $f(x)$ повышает быстродействие алгоритма. Как уже отмечалось в предыдущем разделе, для составных квадратурных формул Ньютона — Котеса при удвоении N (числа внутренних промежутков) в половине возникающих узлов x_k значения функций $f(x_k)$ уже вычислялись ранее и могли быть сохранены, что позволяет сократить объем вычислений вдвое.

В программе реализовано два ограничения сверху на объем вычислений. Во-первых, деление промежутка пополам продолжается не более 30 раз. По достижении этой величины соответствующий интеграл на нем считается вычисленным, а промежуток "принятым", независимо от условия (1.11.3). Число таких промежутков, принятых с нарушением условия (1.11.3), содержится в целой части выходного значения переменной FLAG. Следует признать, что длина каждого такого промежутка крайне мала $(b-a)/2^{30} \approx 10^{-9}(b-a)$, и подобная ситуация, как правило, связана с разрывами подынтегральной функции или ее "зашумлением" вычислительной погрешностью. Во-вторых, вводится ограничение сверху на количество вычислений подынтегральной функции $f(x)$. Если этот предел достигнут, то информация о точке x^*, где возникла трудность, отражена в дробной части выходного значения переменной FLAG (там записана величина $\left(b-x^*\right)/(b-a)$).

В [14] приводится пример использования программы для вычисления интеграла

$$I = \int_0^2 \frac{tg(x)}{x}dx.$$

QUANC8 выдает значение переменной FLAG равное 91.21, что не только обращает внимание на 91 промежуток, принятый с нарушением условия (1.11.3), но и указывает на точку x^*, где встретились затруднения. Определив x^*,

$$\left(b-x^*\right)/(b-a) = 0.21; \quad x^* = b - 0.21(b-a) = 1.58,$$

пользователь вынужден обратить внимание на неинтегрируемую особенность в точке $x = \pi/2 \approx 1.57$.

Программа имеет следующие параметры

```
QUANC8 (FUN, A, B, ABSERR, RELERR, RESULT, ERREST, NOFUN, FLAG)
```

где:

- FUN — имя подпрограммы-функции, вычисляющей значение подынтегральной функции $f(x)$;
- A, B — нижний и верхний пределы интегрирования;
- ABSERR и RELERR — границы абсолютной ε_A и относительной ε_R погрешностей.

Остальные параметры — выходные со следующим смыслом:

- RESULT — значение интеграла, определенное программой;
- ERREST — оценка погрешности, выполненная программой и удовлетворяющая (1.11.3);
- NOFUN — количество вычислений подынтегральной функции $f(x)$, использованных для получения результата;
- FLAG — индикатор надежности результата. Нулевое значение этой переменной отвечает относительной надежности результата, а ненулевое, как уже отмечалось, свидетельствует об отклонениях от нормального хода выполнения программы.

В заключение раздела два вопроса.

Вопрос 10

Предложенная программа *QUANC8* построена на основе формулы Ньютона — Котеса с девятью узлами. Возможно ли построение аналогичной адаптивной программы на основе формулы Симпсона? Если "нет", то почему? Если "да", то в каких частях программы произойдут изменения?

Вопрос 11

Та же группа вопросов, но уже применительно к формуле Гаусса, например, с девятью узлами.

1.12. Численное дифференцирование

Предлагаемая задача ставится следующим образом. Для таблично заданной функции:

x	x_0	x_1	x_2	...	x_m
$f(x)$	$f(x_0)$	$f(x_1)$	$f(x_2)$	...	$f(x_m)$

требуется оценить значения производной функции $f(x)$ в узлах таблицы.

Книга известного американского ученого Р. В. Хемминга "Численные методы" [15] начинается с весьма полезного совета читателю: "Прежде чем решать задачу, подумай, что делать с ее решением". Последуем этой рекомендации.

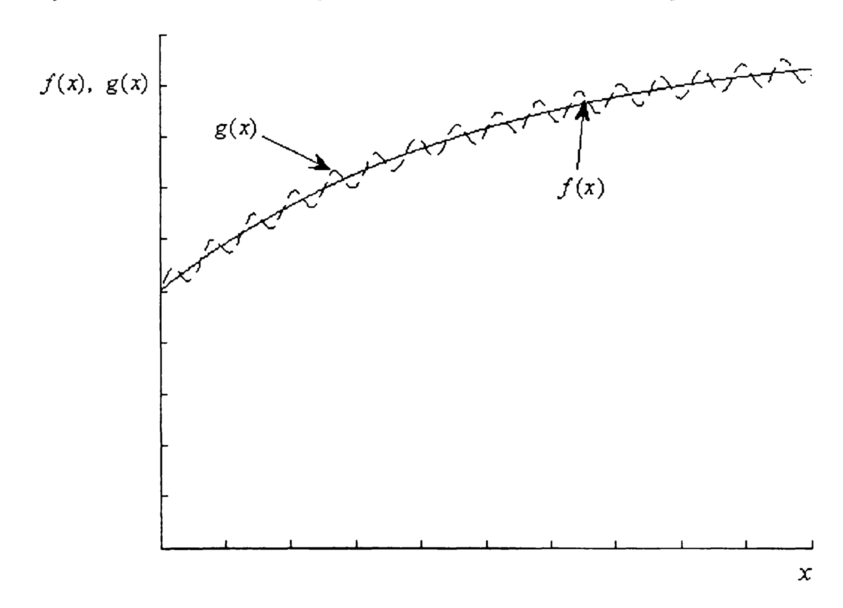


Рис. 1.6. Функция $f(x)$ и ее "зашумленный" вариант

Пусть для функции $f(x)$ выбрана аппроксимирующая функция $g(x)$, которая признана достаточно близкой к исходной по какому-либо заданному критерию, например, по максимуму модуля отклонения (1.1.1). Будут ли близки интегралы от этих функций? Положительный ответ на этот вопрос очевиден. А будут ли близки их производные? На рис. 1.6 отражен вид двух функций: $f(x)$ и ее

"зашумленного" варианта $g(x)$. Сами функции и интегралы от них (т. е. площади под обеими кривыми) весьма близки, в то время как производные совпадают только в нескольких точках, а часто отличаются даже по знаку!

Еще более убедительным является следующий пример. Рассмотрим две функции $f(x)$ и $g(x)$ и их производные:

$$f(x); \qquad g(x) = f(x) + \frac{1}{N}\sin\left(N^2 x\right);$$

$$\frac{df(x)}{dx}; \qquad \frac{dg(x)}{dx} = \frac{df(x)}{dx} + N\cos\left(N^2 x\right).$$

С ростом величины N обе функции становятся по отношению друг к другу все ближе и ближе, а их производные... все дальше и дальше! Таким образом, близость $f(x)$ и $g(x)$ еще не гарантирует близости их производных.

Для того чтобы повысить уверенность в надежном решении задачи численного дифференцирования, на практике делают целый ряд предположений о характере дифференцируемой функции. Шаг таблицы должен быть не слишком велик и согласован с быстротой изменения функции. Сама функция не должна быть измерена со слишком большими погрешностями (т. е. быть "зашумленной"), не должна слишком резко изменяться и т. п. Примеры функций, подобных $g(x)$ на рис. 1.6, должны быть исключены. Таким образом, численное дифференцирование должно выполняться с максимальной осторожностью и сопровождаться предварительным анализом характера изменения функции и погрешности исходных данных. В силу указанных причин с повышением порядка получаемой производной значительно возрастает риск получения отрицательного результата.

Идея, лежащая в основе численного дифференцирования, крайне проста и уже использовалась при получении квадратурных формул (см. (1.9.2) и (1.9.3)). Исходная функция аппроксимируется интерполяционным полиномом

$$f(x) = Q_m(x) + R_m(x),$$

и производная от полинома дает формулу численного дифференцирования

$$\frac{df(x_k)}{dx} \approx \frac{dQ_m(x_k)}{dx},$$

а производная от остаточного члена позволяет оценить погрешность этой операции

$$\varepsilon = \frac{dR_m(x_k)}{dx}.$$

Ограничимся случаем, когда узлы таблицы — равноотстоящие с шагом $h = x_{k+1} - x_k$. Начнем с полинома первой степени, построенного по двум узлам x_k и x_{k+1}:

$$Q_1(x) = \frac{x - x_{k+1}}{x_k - x_{k+1}} f(x_k) + \frac{x - x_k}{x_{k+1} - x_k} f(x_{k+1}).$$

Дифференцируя его и полагая последовательно $x = x_k$ и $x = x_{k+1}$, получаем:

$$\frac{df(x_k)}{dx} \approx \frac{f_{k+1} - f_k}{h}, \qquad (1.12.1)$$

$$\frac{df(x_{k+1})}{dx} \approx \frac{f_{k+1} - f_k}{h}. \qquad (1.12.2)$$

Хотя правые части обоих выражений равны, формулы получились принципиально различными. Выполним аналогичные операции для полинома второй степени:

$$Q_2(x) = \frac{(x - x_{k+1})(x - x_{k+2})}{(x_k - x_{k+1})(x_k - x_{k+2})} f(x_k) + \frac{(x - x_k)(x - x_{k+2})}{(x_{k+1} - x_k)(x_{k+1} - x_{k+2})} f(x_{k+1}) +$$
$$+ \frac{(x - x_k)(x - x_{k+1})}{(x_{k+2} - x_k)(x_{k+1} - x_{k+2})} f(x_{k+2}).$$

Последовательно полагая $x = x_k$, $x = x_{k+1}$ и $x = x_{k+2}$, получаем:

$$\frac{df(x_k)}{dx} \approx \frac{-3f_k + 4f_{k+1} - f_{k+2}}{2h}, \qquad (1.12.3)$$

$$\frac{df(x_{k+1})}{dx} \approx \frac{f_{k+2} - f_k}{2h}, \qquad (1.12.4)$$

$$\frac{df(x_{k+2})}{dx} \approx \frac{3f_{k+2} - 4f_{k+1} + f_k}{2h}. \qquad (1.12.5)$$

Для оценки погрешности всех формул необходимо продифференцировать остаточный член $R_m(x)$. Для полинома первой степени он имеет вид:

$$R_1(x) = \frac{(x - x_k)(x - x_{k+1})}{2!} f''(\eta).$$

Уместно напомнить, что в конце *разд. 1.3* отмечалась зависимость величины η от точки x, где оценивается погрешность, т. е. $\eta = \eta(x)$, и этот факт нужно учитывать при дифференцировании:

$$\frac{dR_1(x)}{dx} = \frac{x - x_{k+1}}{2!} f''(\eta) + \frac{x - x_k}{2!} f''(\eta) + \frac{(x - x_k)(x - x_{k+1})}{2!} f'''(\eta)\eta'(x).$$

Спасает ситуацию то, что погрешность нужно оценивать в узлах интерполирования. Так при $x = x_k$ и $x = x_{k+1}$ два слагаемых из трех в этой формуле обращаются в нуль:

$$\varepsilon_1(x_k) = \frac{dR_1(x_k)}{dx} = -\frac{h}{2} f''(\eta), \tag{1.12.6}$$

$$\varepsilon_1(x_{k+1}) = \frac{dR_1(x_{k+1})}{dx} = \frac{h}{2} f''(\eta). \tag{1.12.7}$$

Выражения (1.12.6) и (1.12.7) задают погрешность численного дифференцирования для формул (1.12.1) и (1.12.2) соответственно. Аналогично продифференцируем погрешность $R_2(x)$:

$$R_2(x) = \frac{(x - x_k)(x - x_{k+1})(x - x_{k+2})}{3!} f'''(\eta),$$

последовательно подставляя в результат $x = x_k$, $x = x_{k+1}$ и $x = x_{k+2}$

$$\varepsilon_2(x_k) = \frac{dR_1(x_k)}{dx} = \frac{h^2}{3} f'''(\eta), \tag{1.12.8}$$

$$\varepsilon_2(x_{k+1}) = \frac{dR_1(x_{k+1})}{dx} = -\frac{h^2}{6} f'''(\eta), \tag{1.12.9}$$

$$\varepsilon_2(x_{k+2}) = \frac{dR_1(x_{k+2})}{dx} = \frac{h^2}{3} f'''(\eta) \tag{1.12.10}$$

и определяя, таким образом, погрешность численного дифференцирования для формул (1.12.3)—(1.12.5) соответственно. Легко заметить, что на меньшую погрешность можно рассчитывать, используя формулу (1.12.4), которая и является наиболее популярной на практике. Формулы же (1.12.3) и (1.12.5) используются для дифференцирования в начале и в конце таблицы соответственно.

Если интерполяционный полином второй степени продифференцировать дважды, то получается простейшая формула для второй производной:

$$\frac{d^2 f(x_{k+1})}{dx^2} \approx \frac{f_{k+2} - 2f_{k+1} + f_k}{h^2}. \qquad (1.12.11)$$

Предложенный подход можно было бы развивать и дальше, привлекая новые узлы интерполирования и дифференцируя полиномы более высоких степеней, но мы ограничимся рассмотренными формулами.

Практически важным является вопрос о выборе шага h для формул численного дифференцирования. Ограничение сверху накладывается величиной погрешности (1.12.6)—(1.12.10), а снизу — точностью задания табличных данных для $f(x)$, что, в свою очередь, определяется, например, результатами эксперимента или ошибками округления (разрядной сеткой компьютера).

1.12.1. Влияние погрешности задания функции на точность

В качестве примера вновь обратимся к простейшей формуле для первой производной (1.12.1). Пусть в ней значение f_{k+1} определено с погрешностью Δ_{k+1}, а значение f_k — с погрешностью Δ_k. Тогда общая погрешность $\varepsilon(h)$ складывается из двух погрешностей

$$\varepsilon(h) = \varepsilon_1(h) + \varepsilon_2(h),$$

первая из которых $\varepsilon_1(h)$ задается формулой (1.1.6) и примерно линейно убывает с уменьшением шага h, а вторая зависит от Δ_k и Δ_{k+1}. Оценивая полную погрешность сверху, получаем

$$|\varepsilon(h)| \le |\varepsilon_1(h)| + |\varepsilon_2(h)| = \frac{h}{2}|f''(\eta)| + \frac{|\Delta_{k+1} - \Delta_k|}{h}. \qquad (1.12.12)$$

Налицо две противоречивые тенденции: с уменьшением h роль первого слагаемого, определяемого погрешностью формулы дифференцирования, уменьшается, а роль второго слагаемого возрастает. Таким образом, недопустимо использование как слишком большого шага h, так и слишком малого. Оптимальное значение шага h_{opt} отвечает ситуации, когда оба слагаемых равны друг другу. Все сказанное и отражает, на первый взгляд, парадоксальный рис. 1.7 оценки сверху для $\varepsilon(h)$.

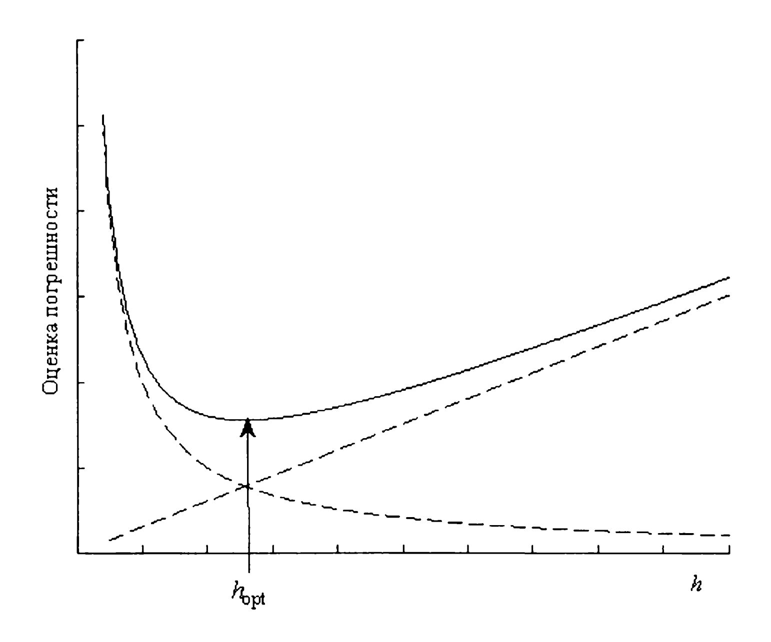


Рис. 1.7. Оценка погрешности (1.12.12)

На практике в точном определении нет необходимости, важно лишь знать о характере зависимости (1.12.12). В заключение напомним, что величины Δ_k, Δ_{k+1} и их вклад в общую погрешность численного дифференцирования могут быть как относительно малы, если определяются разрядной сеткой компьютера, так и весьма велики, если являются следствием погрешности измерения экспериментальных данных.

1.13. Среднеквадратичная аппроксимация функций. Постановка задачи

Критерий интерполирования, предполагающий совпадение исходной и аппроксимирующей функций в узлах таблицы, не является единственным. Обратимся к экспериментальным данным, представленным на рис. 1.8, *а*. Если выполнить по ним интерполяцию, то получится кривая на рис. 1.8, *б*. Маловероятно, чтобы ее вид отвечал исходной зависимости, положенной в основу таблицы. Вероятнее всего, что этой зависимости отвечает кривая, похожая на

рис. 1.8, *в*, а отклонение экспериментальных данных от нее продиктовано сравнительно большой погрешностью измерений. В таком случае целесообразно использовать среднеквадратичный критерий (1.1.3) или (1.1.2), если аппроксимируемая функция задана на $[a,\ b]$ непрерывно.

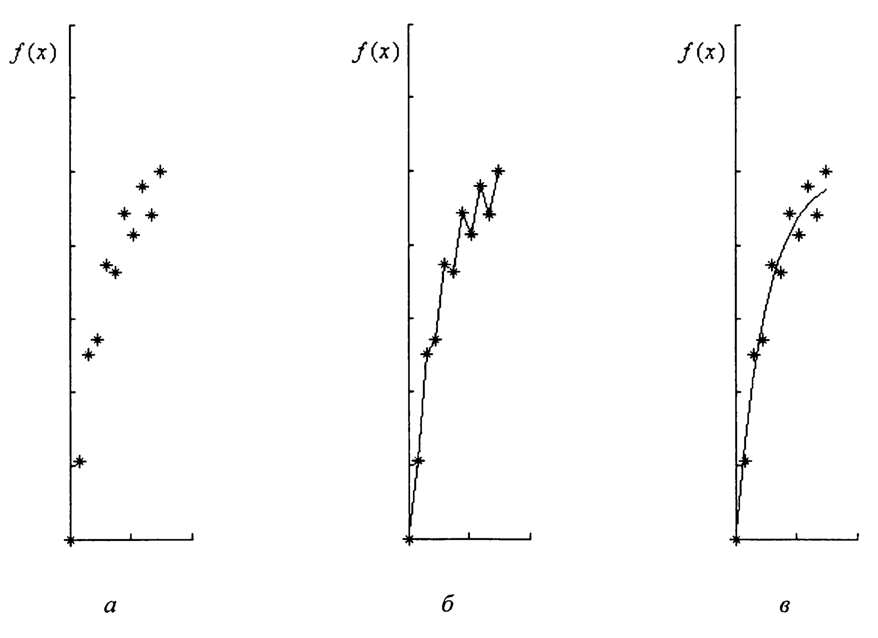


Рис. 1.8. Аппроксимация экспериментальных данных

Для более лаконичной формы записи последующих формул удобно ввести скалярное произведение (f, g) для двух множеств функций *(подробнее см. приложение 2)*.

1. Множество непрерывных вещественных функций на промежутке $[a,\ b]$ образуют линейное пространство, которое при введении в нем скалярного произведения $(f, g) = \int\limits_a^b f(x)g(x)dx$ определяет расстояние между элементами следующим образом:

$$\rho(f,g) = \|f - g\| = \sqrt{(f-g, f-g)} = \sqrt{\int\limits_a^b (f(x) - g(x))^2 dx}.$$

Скалярное произведение может вводиться и в более общей форме, а именно

$$(f,g)=\int_a^b p(x)f(x)g(x)dx,$$

где $p(x)>0$ — весовая функция.

2. Множество функций, заданных на дискретном множестве точек: x_1, x_2, ..., x_N, образуют линейное пространство со скалярным произведением

$$(f,g)=\sum_{i=1}^{N} f(x_i)g(x_i).$$

Расстояние между элементами такого пространства имеет вид:

$$\rho(f,g)=\|f-g\|=\sqrt{(f-g,f-g)}=\sqrt{\sum_{i=1}^{N}(f(x_i)-g(x_i))^2}.$$

И здесь скалярное произведение можно вводить с весовыми коэффициентами $p(x_i)$:

$$(f,g)=\sum_{i=1}^{N} p(x_i)f(x_i)g(x_i).$$

Задача среднеквадратичной аппроксимации функции $f(x)$ для любого из этих двух множеств ставится следующим образом. Требуется подобрать такую аппроксимирующую функцию $Q(x)$, чтобы квадрат расстояния между $Q(x)$ и $f(x)$ был бы минимальным:

$$\rho^2=\|Q(x)-f(x)\|^2=\big(Q(x)-f(x),\ Q(x)-f(x)\big)\to\min.$$

Напомним, что, как уже обсуждалось в *разд. 1.1*, близость функций по среднеквадратичному критерию (1.1.2) еще не гарантирует малой величины их максимальной разности (1.1.1). Малое значение интеграла или суммы для ρ^2 свидетельствует лишь о том, что почти на всем отрезке $[a,\ b]$ значения $f(x)$ и $Q(x)$ мало отличаются друг от друга, хотя в отдельных точках или на небольших отрезках разность их значений может быть значительной.

В качестве иллюстрации рассмотрим функцию $f(x)$, график которой имеет узкий зубец высоты h с основанием $1/h^3$ и функцию $Q(x)\equiv 0$ (рис. 1.9).

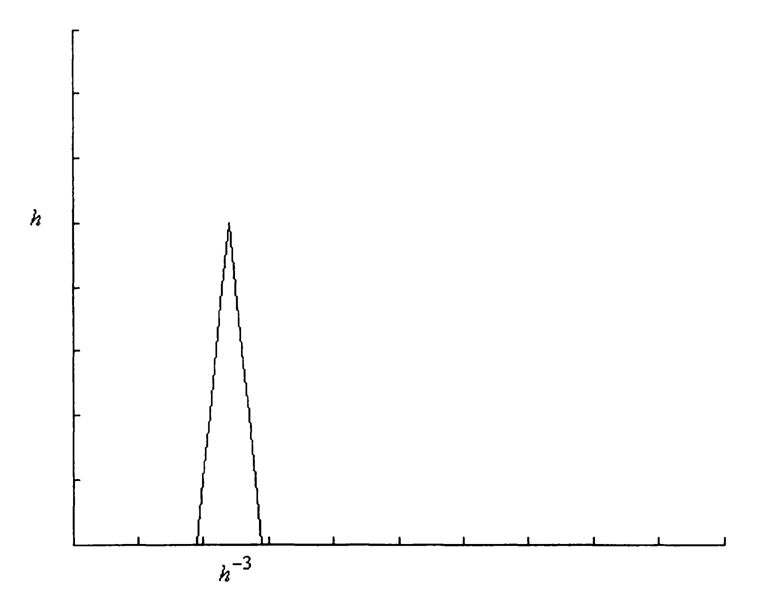


Рис. 1.9. Пример функции с малой величиной ρ^2

Нетрудно вычислить:

$$\rho(f,Q)=\|f-Q\|=\sqrt{(f-Q,f-Q)}=\sqrt{\int_a^b (f(x)-Q(x))^2\,dx}=$$

$$=\sqrt{\int_a^b f^2(x)dx}<\sqrt{\frac{h^2}{h^3}}=\frac{1}{\sqrt{h}}.$$

За счет выбора h среднеквадратичное расстояние $\rho(f,Q)$ можно сделать сколь угодно малым, а величину $\max\limits_{[a,b]}|f-Q|=h$ сколь угодно большой.

Проблема выбора аппроксимирующей функции решается так же, как и при интерполяции: $Q(x)$ выбирается в виде обобщенного многочлена (1.2.1)

$$Q_m(x)=a_0\varphi_0(x)+a_1\varphi_1(x)+\ldots+a_m\varphi_m(x)=\sum_{k=0}^{m}\varphi_k(x), \qquad (1.13.1)$$

где $\{\varphi_k\}$ — заданный набор линейно независимых функций, а коэффициенты a_k подлежат определению.

1.13.1. Дискретный случай. Весовые коэффициенты

Функция $f(x)$ задается на дискретном множестве точек следующей таблицей:

x	x_1	x_2	x_3	...	x_N
$f(x)$	$f(x_1)$	$f(x_2)$	$f(x_3)$		$f(x_N)$

а ее аппроксимация $Q_m(x)$ выбирается в виде обобщенного многочлена (1.13.1). Коэффициенты a_k выбираются из условия минимума величины ρ^2:

$$\rho^2 = \left(Q(x) - f(x),\ Q(x) - f(x)\right) = \sum_{i=1}^{N}\left(Q(x_i) - f(x_i)\right)^2 \to \min. \quad (1.13.2)$$

Рассмотрим три варианта соотношений чисел N и m.

- $N = m+1$. Число коэффициентов a_k равно числу точек таблицы, решение задачи единственное, и им является интерполяционный полином, проходящий через все точки. Минимальное значение ρ^2 оказывается нулевым.
- $N < m+1$. Минимальное значение ρ^2 также равно нулю, но задача имеет бесконечное множество решений.
- $N > m+1$. Это типичный случай среднеквадратичной аппроксимации. Более того, на практике часто $N \gg m+1$. Минимальное значение ρ^2 оказывается уже, как правило, ненулевым, а задача имеет единственное решение. В данном разделе ограничимся этим третьим вариантом, т. к. первый был уже рассмотрен, а ко второму вернемся в *главе 6*.

Записываем необходимое условие экстремума:

$$\frac{\partial \rho^2}{\partial a_k} = 0, \qquad k = 0,\ 1,\ 2, \ldots, m$$

и выполняем операцию дифференцирования:

$$\frac{\partial \rho^2}{\partial a_k} = \frac{\partial}{\partial a_k}(Q - f, Q - f) = \left(\frac{\partial Q}{\partial a_k}, Q - f\right) + \left(Q - f, \frac{\partial Q}{\partial a_k}\right) = 2\left(\frac{\partial Q}{\partial a_i}, Q - f\right) = 0.$$

Подставляя в получившуюся формулу выражение для $Q_m(x)$, получаем систему линейных алгебраических уравнений относительно a_k:

$$(\varphi_0,\varphi_k)a_0+(\varphi_1,\varphi_k)a_1+\ldots+(\varphi_m,\varphi_k)a_m=(f,\varphi_k),\quad k=0,1,\ldots,m. \qquad (1.13.3)$$

Линейная независимость системы базисных функций $\{\varphi_k(x)\}$ на заданном множестве точек обеспечивает ненулевой определитель системы

$$\begin{vmatrix} (\varphi_0,\varphi_0) & (\varphi_1,\varphi_0) & \ldots & (\varphi_m,\varphi_0) \\ (\varphi_0,\varphi_1) & (\varphi_1,\varphi_1) & \ldots & (\varphi_m,\varphi_1) \\ \ldots & \ldots & \ldots & \ldots \\ (\varphi_0,\varphi_m) & (\varphi_1,\varphi_m) & \ldots & (\varphi_m,\varphi_m) \end{vmatrix}, \qquad (1.13.4)$$

и задача имеет единственное решение. Самой популярной является аппроксимация полиномами, когда $\varphi_k(x)=x^k$, а система (1.13.3) приобретает вид

$$\left(\sum_{i=1}^{N}1\right)a_0+\left(\sum_{i=1}^{N}x_i\right)a_1+\ldots+\left(\sum_{i=1}^{N}x_i^{\,m}\right)a_m=\left(\sum_{i=1}^{N}f(x_i)\right),$$

$$\left(\sum_{i=1}^{N}x_i\right)a_0+\left(\sum_{i=1}^{N}x_i^{\,2}\right)a_1+\ldots+\left(\sum_{i=1}^{N}x_i^{\,m+1}\right)a_m=\left(\sum_{i=1}^{N}f(x_i)x_i\right), \qquad (1.13.5)$$

$$\ldots,$$

$$\left(\sum_{i=1}^{N}x_i^{\,m}\right)a_0+\left(\sum_{i=1}^{N}x_i^{\,m+1}\right)a_1+\ldots+\left(\sum_{i=1}^{N}x_i^{\,2m}\right)a_m=\left(\sum_{i=1}^{N}f(x_i)x_i^{\,m}\right).$$

На практике с ростом m определитель системы быстро уменьшается, а матрица становится *плохо обусловленной* (подробнее об этом явлении *см. в главе 2*).

При выполнении среднеквадратичной аппроксимации (другое название — "метод наименьших квадратов") возможна ситуация, когда исходные данные имеют различную точность. Если к каким-либо экспериментальным значениям доверие выше, т. е. они являются более надежными по сравнению с другими, это может быть учтено введением в критерий (1.13.2) положительных весовых коэффициентов p_i:

$$\rho^2=\big(Q(x)-f(x),\ Q(x)-f(x)\big)=\sum_{i=1}^{N}p_i\big(Q(x_i)-f(x_i)\big)^2\to\min \qquad (1.13.6)$$

Вопрос 12

Закончите следующую фразу: "Для тех точек, степень доверия которым выше и к которым аппроксимирующую кривую желательно провести ближе, чем к другим точкам, весовые коэффициенты следует задавать ...". (Больше или меньше?)

Конечно, больше. Величину ρ^2 можно трактовать как своеобразную "функцию штрафа". За отклонение $Q_m(x)$ от $f(x)$ в точке x_i к значению ρ^2 добавляется слагаемое ("штраф") $\left(Q(x_i)-f(x_i)\right)^2$ тем большее, чем больше это отклонение. Если какая-то точка является более приоритетной и к ней аппроксимирующую кривую желательно провести ближе, с помощью весового коэффициента за отклонение в этой точке "штраф" должен быть увеличен.

На практике положительные весовые коэффициенты p_i часто задают так, чтобы их сумма была равна, например, 1 или 100. Последнее часто удобно, но не обязательно. Если все коэффициенты умножить на одно и то же число, то, хотя ρ^2 и изменится, решение задачи останется прежним. Важными являются отношения p_i друг к другу.

1.13.2. Непрерывный случай. Понятие ортогональности

Теперь обратимся к варианту непрерывного задания $f(x)$ на $[a,\ b]$. Выражение (1.13.3) предыдущего раздела сохранит свой внешний вид, только скалярные произведения и критерий ρ^2 будут записываться не через суммы, а через интегралы

$$(\varphi_k, \varphi_i) = \int_a^b \varphi_k(x)\varphi_i(x)dx,$$

$$\rho^2 = \left(Q(x)-f(x),\ Q(x)-f(x)\right) = \int_a^b \left(Q(x)-f(x)\right)^2 dx \to \min. \quad (1.13.7)$$

Определитель (1.13.4) оказывается определителем Грама и, как известно из алгебры, отличен от нуля, если базисные функции линейно независимы, что

имеет место. Система (1.13.3) имеет единственное решение, и полином, доставляющий минимум ρ^2, существует.

В качестве примера обратимся к аппроксимации полиномами $\varphi_k(x) = x^k$ на промежутке $[0,\ 1]$. Вычисление определителя Грама дает следующий результат:

$$G_m = \begin{vmatrix} 1 & 1/2 & 1/3 & \dots & 1/(m+1) \\ 1/2 & 1/3 & 1/4 & \dots & 1/(m+2) \\ 1/3 & 1/4 & 1/5 & \dots & 1/(m+3) \\ \dots & \dots & \dots & \dots & \dots \\ 1/(m+1) & 1/(m+2) & 1/(m+3) & \dots & 1/(2m+1) \end{vmatrix}. \qquad (1.13.8)$$

Этот определитель называется *определителем Гильберта*, а соответствующая ему матрица — *матрицей Гильберта*. Она, как и матрица системы (1.13.5), оказывается плохо обусловленной и характеризуется крайне неприятным свойством: весьма малые изменения ее элементов приводят к сильному изменению решения системы. Определитель Гильберта стремительно уменьшается с ростом порядка системы:

$$\begin{array}{lcccccccc} m: & 1 & 2 & 3 & 4 & \dots & 8 & 9 \\ G_m: & 1 & 8.3\times10^{-2} & 4.6\times10^{-4} & 1.7\times10^{-7} & \dots & 2.7\times10^{-33} & 9.7\times10^{-43}. \end{array}$$

Аналогично дискретному случаю критерий ρ^2 может быть обобщен введением положительной весовой функции $p(x)$. Как и ранее, все формулы сохранят свой внешний вид, а скалярное произведение запишется следующим образом:

$$(\varphi_k, \varphi_i) = \int_a^b p(x)\varphi_k(x)\varphi_i(x)dx.$$

Обратимся вновь к системе (1.13.3). Многие проблемы ее решения можно обойти, если вместо произвольных линейно независимых функций $\{\varphi_k(x)\}$ воспользоваться ортогональными функциями $\{g_k(x)\}$ *(см. приложение 2)*. Напомним, что последовательность функций $\{g_k(x)\}$ является ортогональной на промежутке $[a, b]$ с весом $p(x)$, если $(g_k, g_i) = 0$, т. е. выполняются следующие условия:

$$(g_k, g_i) = \int_a^b p(x)g_k(x)g_i(x)dx = \begin{cases} 0, & \text{если } i \neq k, \\ A > 0, & \text{если } i = k. \end{cases} \qquad (1.13.9)$$

Если в дополнение $A=1$, то такие функции называются *ортонормированными* (т. е. ортогональными и нормированными). Для ортогональных функций матрица системы (1.13.3) оказывается диагональной, и каждое уравнение дает готовое выражение для коэффициента a_k:

$$a_k = \frac{(f, g_k)}{(g_k, g_k)} = \frac{\int_a^b p(x) f(x) g_k(x) dx}{\int_a^b p(x) {g_k}^2(x) dx}. \quad (1.13.10)$$

Эти коэффициенты носят название *коэффициентов Фурье* функции $f(x)$ по ортогональной системе функций $g_0(x)$, $g_1(x)$, $g_2(x)$, ..., $g_m(x)$.

На первый взгляд, использование ортогональных функций снимает все проблемы, связанные с решением системы (1.13.3). Однако ряд вычислительных проблем остается. Они связаны с вычислением интеграла в числителе (1.13.19).

Аппроксимирующий полином можно трактовать как разложение исходной функции $f(x)$ по ортогональному базису $g_0(x)$, $g_1(x)$, $g_2(x)$, ..., $g_m(x)$:

$$f(x) \approx a_0 g_0(x) + a_1 g_1(x) + ... + a_m g_m(x).$$

Начиная с некоторого момента, при добавлении новых базисных функций $g_k(x)$ их вклад становится невелик по сравнению с уже построенным разложением, т. е. коэффициенты a_k оказываются малыми по абсолютной величине. Нахождение близкого к нулю значения определенного интеграла сопряжено с трудностями как при аналитическом, так и при приближенном вычислении. Если подынтегральная функция принимает внутри промежутка достаточно большие по модулю значения, то неизбежно произойдет потеря (часто довольно большого числа) верных значащих цифр. В аналитическом случае это будет на завершающем этапе вычисления разности первообразной на верхнем и нижнем пределах, а в приближенном варианте — при использовании квадратурных формул.

Если исходный набор функций $\{\varphi_k(x)\}$ не является ортогональным, то, используя процедуру Грама — Шмидта *(см. приложение 2)*, его можно сделать таковым, построив функции $\{g_k(x)\}$, которые не только станут ортогональными, но и будут линейной комбинацией функций $\{\varphi_k(x)\}$. Аппроксимация, таким образом, будет выполняться в том же классе функций. Одним

3 Зак. 3418

из наиболее популярных естественных ортогональных базисов являются тригонометрические функции, т. е.

$$1,\ \cos x,\ \sin x,\ \cos 2x,\ \sin 2x,\ \ldots,\ \cos mx,\ \sin mx,\ \ldots$$

Изучаемые в курсе математического анализа ряды Фурье в рассматриваемом нами аспекте являются аппроксимацией функций посредством этих рядов. Еще более популярными являются ортогональные полиномы, примеры которых рассматриваются далее.

1.13.3. Ортогональные полиномы и их свойства

Неотъемлемыми атрибутами понятия ортогональности являются промежуток интегрирования и весовая функция. Проблема различных промежутков, возникающих на практике, решается легко. Полиномы для стандартных промежутков (обычно это $[-1,\ 1]$ или $[0,\ 1]$) приводятся в справочниках и учебниках, а к произвольному промежутку переходят обычной заменой переменных. Примером является следующая замена:

$$x = \frac{a+b}{2} + \frac{b-a}{2}t, \quad x \in [a, b], \quad t \in [-1, 1].$$

В приводимых примерах остановимся на стандартном промежутке $[-1, +1]$. Тогда главной отличительной особенностью различных полиномов будет весовая функция.

Ортогональные полиномы Лежандра

Для этих полиномов весовая функция имеет популярный вид: $p(x) \equiv 1$, и сами они могут быть вычислены по формуле:

$$L_n(x) = \frac{(-1)^n}{n!2^n}\frac{d^n}{dx^n}\left[\left(1-x^2\right)^n\right], \quad x \in [-1, 1]. \qquad (1.13.11)$$

Легко заметить, что $L_0(x) = 1$, $L_1(x) = x$, и имеет место следующее рекуррентное соотношение:

$$(n+1)L_{n+1}(x) - (2n+1)xL_n(x) + nL_{n-1}(x) = 0. \qquad (1.13.12)$$

Тогда для последующих полиномов легко получаем:

$$L_2(x)=\frac{3x^2-1}{2},\quad L_3(x)=\frac{5x^3-3x}{2},\quad L_4(x)=\frac{35x^4-30x^2+3}{8},\ \ldots$$

В такой форме полиномы Лежандра ортогональны на $[-1,+1]$, но не нормированы. Квадрат их нормы имеет вид

$$(L_n,L_n)=\int_{-1}^{1}L_n^2(x)dx=\frac{2}{2n+1},$$

а графики первых четырех полиномов представлены на рис. 1.10, *а*.

Ортогональные полиномы Чебышева

Для этих полиномов весовая функция выглядит следующим образом: $p(x)=\frac{1}{\sqrt{1-x^2}}$. При $x\in[-1,\ 1]$ они могут быть вычислены по формуле

$$T_n(x)=\cos(n\cdot\arccos(x));\quad T_0(x)=1,\quad T_1(x)=x. \tag{1.13.13}$$

Как и для полиномов Лежандра, здесь имеет место рекуррентное соотношение

$$T_{n+1}(x)=2xT_n(x)-T_{n-1}(x), \tag{1.13.14}$$

на основе которого легко получаются последующие полиномы:

$$T_2(x)=2x^2-1,\quad T_3(x)=4x^3-3x,\quad T_4(x)=8x^4-8x^2+1,\ \ldots$$

В справедливости (1.13.14) легко убедиться. Подставляя $\varphi=\arccos(x)$ в очевидное соотношение

$$\cos(n+1)\varphi+\cos(n-1)\varphi=2\cos\varphi\cos(n\varphi),$$

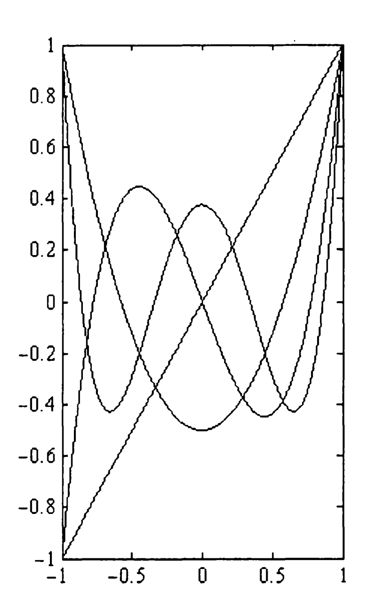
приходим к (1.13.14). Полиномы Чебышева могут быть представлены и в ином виде, отличном от (1.13.13):

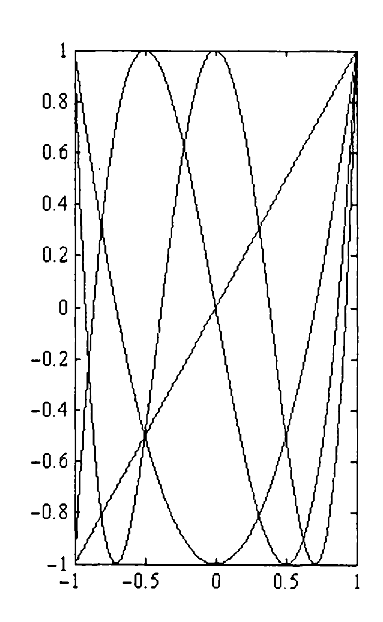
$$T_n(x)=\frac{(-2)^n n!}{(2n)!}\sqrt{1-x^2}\,\frac{d^n}{dx^n}\left[\left(1-x^2\right)^{n-1/2}\right]. \tag{1.13.15}$$

Как и полиномы Лежандра, полиномы Чебышева в форме (1.13.13) ортогональны, но не нормированы. Квадраты их нормы имеют вид

$$(T_0,T_0)=\pi\,,\ (T_n,T_n)=\int_{-1}^{1}\frac{T_n^2(x)}{\sqrt{1-x^2}}dx=\frac{\pi}{2},\qquad n>0,$$

а графики первых четырех полиномов представлены на рис. 1.10, *б*.





а *б*

Рис. 1.10. Полиномы: *а* — Лежандра; *б* — Чебышева

Ортогональные полиномы могут быть построены и на дискретном множестве точек. Пусть независимая переменная принимает значения 0, 1, 2, ..., m; а весовая функция $p(x)\equiv 1$. Полиномы Чебышева степени n задаются формулой:

$$P_{nm}(x)=\sum_{k=0}^{n}\frac{(-1)^k C_n^k C_{n+k}^k}{m^{[k]}}x^{[k]},\quad n=0,1,\ldots,m.$$

Здесь $m^{[k]}$ и $x^{[k]}$ — факториальные многочлены:

$$m^{[k]}=m(m-1)\ldots(m-k+1);\qquad x^{[k]}=x(x-1)\ldots(x-k+1).$$

Запишем несколько первых полиномов:

$$P_{0m}(x) = 1, \quad P_{1m}(x) = 1 - 2\frac{x}{m}, \quad P_{2m}(x) = 1 - 6\frac{x}{m} + 6\frac{x(x-1)}{m(m-1)},$$

$$P_{3m}(x) = 1 - 12\frac{x}{m} + 30\frac{x(x-1)}{m(m-1)} - 20\frac{x(x-1)(x-2)}{m(m-1)(m-2)}, \ \ldots$$

Если, например, $m = 5$, т. е. множество рассматриваемых точек $\{0, 1, 2, 3, 4, 5\}$, то ортогональными на этом множестве точек будут полиномы:

$$P_{05}(x) = 1, \quad P_{15}(x) = 2x - 5, \quad P_{25}(x) = 3x^2 - 15x + 10,$$

$$P_{35}(x) = 10x^3 - 75x^2 + 127x - 30, \ \ldots$$

Убедиться в ортогональности этих полиномов можно, вычислив скалярные произведения:

$$\left(P_{im}(x), P_{jm}(x)\right) = \begin{cases} 0, & i \neq j, \\ \dfrac{(i+m+1)^{[i+1]}}{m^{[i]}(2i+1)}, & i = j. \end{cases}$$

Некоторые свойства ортогональных полиномов

Начнем с теоремы существования и единственности ортогональных полиномов, которую приведем без доказательства.

Теорема 1. Для всякой весовой функции $p(x)$, удовлетворяющей условиям:

- $p(x) \geq 0$ (неотрицательность);
- $\int_a^b p(x)dx < \infty$ (интегрируемость);
- $\int_a^b p(x)dx > 0$ (положительность интеграла)

существует единственная последовательность полиномов $\{S_k(x)\}$, где $k = 1, 2, \ldots, n, \ldots$, имеющих положительный старший коэффициент и удовлетворяющих условию ортонормированности.

Теорема 2. Ортогональный полином $S_m(x)$ ортогонален произвольному полиному $R_k(x)$ меньшей степени:

$$\int_a^b p(x)S_m(x)R_k(x)dx = 0, \quad m > k.$$

Доказательство. Разделим $R_k(x)$ на $S_k(x)$.

$$R_k(x) = a_k S_k(x) + R_{k-1}(x),$$

где a_k — частное от деления старших коэффициентов $R_k(x)$ и $S_k(x)$. Далее разделим $R_{k-1}(x)$ на $S_{k-1}(x)$:

$$R_{k-1}(x) = a_{k-1}S_{k-1}(x) + R_{k-2}(x).$$

Продолжая этот процесс деления, получим

$$R_k(x) = a_k S_k(x) + R_{k-1}(x) = a_k S_k(x) + a_{k-1}S_{k-1}(x) + R_{k-2}(x) = \dots$$
$$= a_k S_k(x) + a_{k-1}S_{k-1}(x) + \dots + a_0 S_0(x).$$

Это разложение полинома $R_k(x)$ по ортогональным полиномам $S_i(x)$. Умножим обе части равенства на $S_m(x)$ ($m > k$) и проинтегрируем с весом $p(x)$ на промежутке ортогональности:

$$\int_a^b p(x)R_k(x)S_m(x)dx = a_k \int_a^b p(x)S_k(x)S_m(x)dx + \dots + a_0 \int_a^b p(x)S_0(x)S_m(x)dx.$$

Все интегралы в правой части равны нулю, и теорема доказана.

Теорема 3. Все m корней ортогонального полинома $S_m(x)$ являются вещественными и лежат на промежутке ортогональности $[a, b]$.

Доказательство. Разобьем процедуру на три части в зависимости от того, каковыми могут быть корни. Доказывать будем от противного.

Покажем, что нет вещественных корней кратности большей, чем 1. Предположим, что корень x_1 имеет четную кратность q. Тогда

$$S_m(x) = (x - x_1)^q R_{m-q}(x).$$

Умножим обе части равенства на $R_{m-q}(x)$ и проинтегрируем с весом $p(x)$ на промежутке ортогональности:

$$\int_a^b p(x)S_m(x)R_{m-q}(x)dx = \int_a^b p(x)(x - x_1)^q R_{m-q}^2(x)dx.$$

Интеграл слева равен нулю по теореме 2, а выражение справа содержит под интегралом неотрицательную функцию, что приводит к противоречию. Если q принимает нечетное значение, большее единицы, то умножать $S_m(x)$ следует на $(x - x_1) \cdot R_{m-q}(x)$ с аналогичными рассуждениями.

Покажем, что нет комплексных корней. Пусть трехчлен $x^2 + ax + b$ определяет пару комплексных корней. Тогда

$$S_m(x) = \left(x^2 + ax + b\right)R_{m-2}(x).$$

Вновь умножаем обе части равенства на $p(x)R_{m-2}(x)$, интегрируем на $[a, b]$ и получаем

$$\int_a^b p(x)S_m(x)R_{m-2}(x)dx = \int_a^b p(x)(x^2 + ax + bx)R_{m-2}^2(x)dx.$$

Снова противоречие, т. к. интеграл слева равен нулю, а справа под интегралом стоит неотрицательная функция.

Наконец, пусть x_1 — корень вне $[a, b]$. Тогда

$$S_m(x) = (x - x_1)R_{m-1}(x).$$

Выполнив аналогичные операции,

$$\int_a^b p(x)S_m(x)R_{m-1}(x)dx = \int_a^b p(x)(x - x_1)R_{m-1}^2(x)dx,$$

вновь приходим к противоречию, т. к. подынтегральная функция справа знакопостоянна.

Теорема 4. Для любых трех соседних ортогональных полиномов справедлива рекуррентная формула (разностное уравнение второго порядка):

$$a_{m+1}S_{m+1}(x) + (a_m - x)S_m(x) + a_{m-1}S_{m-1}(x) = 0,$$

где a_k — постоянные (не зависящие от x) коэффициенты.

Доказательство. Выпишем разложение $xS_m(x)$ по ортогональным полиномам, как это делалось в теореме 2:

$$xS_m(x) = a_{m+1}S_{m+1}(x) + a_mS_m(x) + a_{m-1}S_{m-1}(x) + \ldots + a_0S_0(x). \qquad (1.13.16)$$

Умножая равенство последовательно на $p(x)S_k(x)$, где $k=0,\ 1,\ ...,\ m-2$ и интегрируя на промежутке $[a,\ b]$, без труда установим, что $a_0,\ a_1,\ ...,\ a_{m-2}$ все равны нулю. Действительно,

$$\int_a^b p(x)S_k(x)xS_m(x)dx = a_{m+1}\int_a^b p(x)S_{m+1}(x)S_k(x)dx +$$
$$+\ a_m\int_a^b p(x)S_m(x)S_k(x)dx + ... + a_k\int_a^b p(x)S_k(x)S_k(x)dx + ... +$$
$$+\ a_0\int_a^b p(x)S_0(x)S_k(x)dx.$$

Интеграл слева всегда равен нулю, а равенство нулю правой части можно обеспечить, лишь положив $a_k = 0$. Тогда от исходного выражения остается

$$xS_m(x) = a_{m+1}S_{m+1}(x) + a_mS_m(x) + a_{m-1}S_{m-1}(x)$$

или

$$a_{m+1}S_{m+1}(x) + (a_m - x)S_m(x) + a_{m-1}S_{m-1}(x) = 0.$$

Еще одно свойство приведем без доказательства.

Теорема 5. Пусть полином $S_m(x)$ имеет корни $x_1,\ x_2,\ ...,\ x_m$, а полином $S_{m+1}(x)$ — корни $\zeta_1,\ \zeta_2,\ ...,\ \zeta_m,\ \zeta_{m+1}$. Тогда справедливо неравенство

$$a < \zeta_1 < x_1 < \zeta_2 < x_2 < ... < \zeta_m < x_m < \zeta_{m+1} < b.$$

Из рассмотренных свойств весьма конструктивным выглядит теорема 4, которая позволяет строить любые ортогональные полиномы, если известны два первых из них. Формулы (1.13.12) и (1.13.14) являются примерами выражения (1.13.16).

В заключение данного раздела вернемся к двум ранее уже объявленным результатам. В *разд. 1.4* при выборе узлов интерполирования отмечалось, что оптимальный их выбор отвечает нулям ортогональных полиномов Чебышева. Опуская строгое доказательство этого факта, отметим лишь, что, как это следует из выражения (1.13.13), график модуля $T_n(x)$ $(|T_n(x)| = |\cos(n \arccos(\mathrm{x}))|)$ и порождает одинаковые по высоте "колокольчики", представленные на рис. 1.3.

Второе замечание относится к квадратурным формулам Гаусса. Как отмечалось, узлы x_k этих формул являются нулями ортогональных полиномов Лежандра. В качестве упражнения установим справедливость этого факта. При взятии интеграла вместо исходной функции воспользуемся полиномом Эрмита (1.7.2) степени $2m+1$. При этом в каждом узле x_0, x_1, ..., x_m заданы функция $f(x_k)$ и ее производная $f'(x_k)$:

$$H_{2m+1}(x)=\sum_{k=0}^{m}\left[\frac{\omega(x)}{(x-x_k)\omega'(x_k)}\right]^2\left[f(x_k)\left(1-\frac{\omega''(x_k)}{\omega'(x_k)}(x-x_k)\right)+f'(x_k)(x-x_k)\right].$$

Подставим полином в интеграл, меняя порядок интегрирования и суммирования:

$$\int_a^b f(x)dx\approx\int_a^b H_{2m+1}(x)dx=\sum_{k=0}^{m}\int_a^b\left[\frac{\omega(x)}{(x-x_k)\omega'(x_k)}\right]^2 dx f(x_k)+$$

$$+\sum_{k=0}^{m}\left(f'(x_k)-\frac{\omega''(x_k)}{\omega'(x_k)}f(x_k)\right)-\int_a^b\left[\frac{\omega(x)}{(x-x_k)\omega'(x_k)}\right]^2(x-x_k)dx=$$

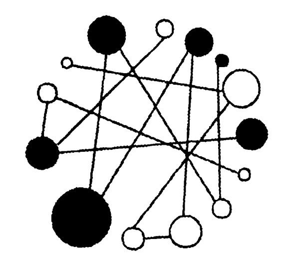
$$=\sum_{k=0}^{m}A_k f(x_k)+\sum_{k=0}^{m}\left(f'(x_k)-\frac{\omega''(x_k)}{\omega'(x_k)}f(x_k)\right)-\int_a^b\left[\frac{\omega(x)}{(x-x_k)\omega'(x_k)}\right]^2(x-x_k)dx.$$

В качестве узлов интерполирования выберем нули ортогональных полиномов Лежандра и сделаем, таким образом, функцию $\omega(x)$ полиномом Лежандра. Первое слагаемое в правой части является квадратурной формулой Гаусса, а интеграл во втором слагаемом обращается в нуль. Действительно, для этого интеграла имеем

$$\int_a^b\left[\frac{\omega(x)}{(x-x_k)\omega'(x_k)}\right]^2(x-x_k)dx=\left(\frac{1}{\omega'(x_k)}\right)^2\int_a^b\omega(x)\frac{\omega(x)}{(x-x_k)}dx.$$

Функция $\omega(x)$ — полином Лежандра, а $\dfrac{\omega(x)}{(x-x_k)}$ — полином на степень меньше, и по теореме 2 интеграл равен нулю. Погрешность квадратурной формулы при этом может быть получена интегрированием остаточного члена (1.7.3) полинома Эрмита.

ГЛАВА 2



Задачи линейной алгебры

Традиционно к задачам линейной алгебры относят две большие проблемы: решение линейных алгебраических систем

$$\mathbf{Ax} = \mathbf{b} \tag{2.0.1}$$

и нахождение собственных значений и собственных векторов матрицы $\mathbf{A}$. К решению (2.0.1), где $\mathbf{A}$ — невырожденная квадратная матрица порядка $m \times m$, $\mathbf{b}$ — заданный вектор-столбец и $\mathbf{x}$ — искомый вектор-столбец, с n компонентами каждый, примыкают задачи:

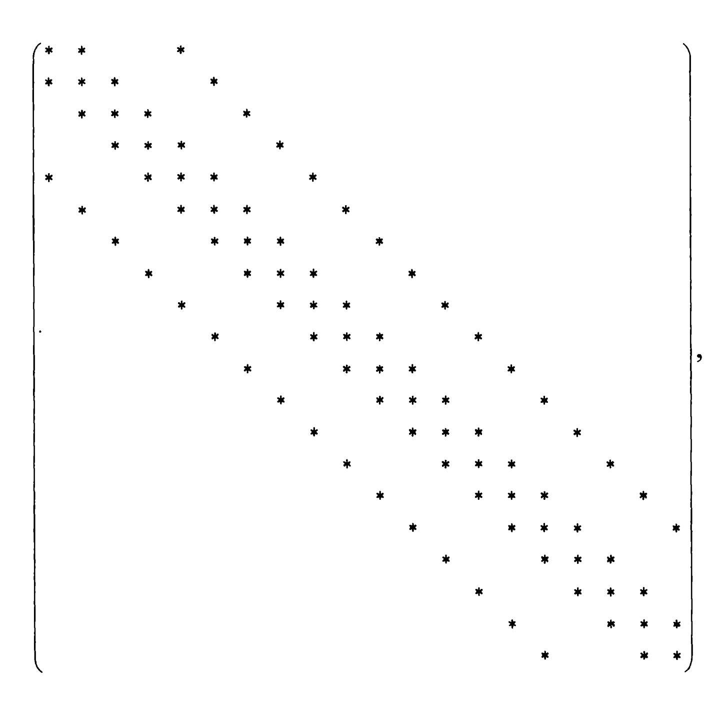
- решение (2.0.1) с прямоугольными матрицами $\mathbf{X}$ и $\mathbf{B}$ ($\mathbf{AX} = \mathbf{B}$);
- нахождение обратной матрицы $\mathbf{A}^{-1}$;
- вычисление определителя матрицы $\mathbf{A}$.

Вторая проблема связана с нахождением собственных значений, удовлетворяющих уравнению $\det(\mathbf{A} - \lambda\mathbf{E}) = 0$, и собственных векторов, являющихся решением уравнения $\mathbf{Ax} = \lambda\mathbf{x}$ для всех λ. Если матрица $\mathbf{A}$ симметрическая, задача значительно упрощается, т. к. в этом случае собственные значения вещественные. Иногда возникает обобщенная задача на собственные значения $\mathbf{Ax} = \lambda\mathbf{Cx}$, где $\mathbf{A}$ — вещественная симметрическая матрица, $\mathbf{C}$ — вещественная симметрическая положительно определенная матрица.

Еще две задачи можно отнести к вычислительным задачам линейной алгебры. В первой из них требуется найти такой вектор $\mathbf{x}$, при котором норма вектора невязки $\mathbf{r} = \mathbf{Cx} - \mathbf{d}$ минимальна ($\|\mathbf{Cx} - \mathbf{d}\| \to \min$). Вектор $\mathbf{x}$ является решением по методу наименьших квадратов обычно несовместной системы $\mathbf{Cx} = \mathbf{d}$. Здесь матрица $\mathbf{C}$ — прямоугольная с числом строк, превышающим число столбцов, а у вектора $\mathbf{d}$ столько элементов, сколько строк у матрицы $\mathbf{C}$. К этой проблеме мы вернемся в *главе 6*.

Во второй задаче для заданной матрицы $\mathbf{A}$, векторов $\mathbf{b}$ и $\mathbf{c}$ требуется найти такой вектор $\mathbf{x}$, чтобы выполнялись неравенства $\mathbf{Ax} \geq \mathbf{b}$ и $\mathbf{x} \geq 0$, а скалярное произведение $(\mathbf{c}, \mathbf{x})$ было бы минимальным. Матрица $\mathbf{A}$ является прямоугольной с числом строк, превышающим число столбцов. Эта задача требует специальных подходов и изучается в разделах линейного программирования.

При решении указанных выше задач встречаются матрицы различных типов. Если число ненулевых элементов матрицы порядка числа ее строк, т. е. большинство элементов являются нулевыми, то такую матрицу называют *разреженной*. При этом ненулевые элементы либо хранятся с использованием специальных типов данных, либо генерируются по мере необходимости. В других случаях, когда все a_{ij} хранятся в оперативной памяти компьютера, матрица называется *хранимой*. Часто разреженные матрицы возникают при конечно-разностных аппроксимациях уравнений с частными производными. Примером разреженной матрицы может служить *ленточная* матрица



отличные от нуля элементы которой располагаются вблизи главной диагонали: $a_{ij} = 0$ для i, j, удовлетворяющих условию $|i-j| > s$, где $s \ll m$. Число $2s+1$ называют *шириной ленты*, оно задает количество диагоналей с ненулевыми элементами.

2.1. Обусловленность матриц

Численное решение линейных алгебраических систем подвержено влиянию нескольких источников ошибок. Два из них традиционны и очевидны: ограниченность разрядной сетки компьютера и погрешность представления исходных данных. Единственное решение система (2.0.1) имеет при $\det(\mathbf{A}) \neq 0$.

Если изменение элементов в пределах точности задания данных с наложением на них ошибок округления приведет к нулевому определителю, получить единственное решение не удастся. Для анализа возникшей ситуации введем понятие обусловленности матриц.

Первоначально оно появилось, как чисто качественная характеристика чувствительности элементов обратной матрицы $\mathbf{A}^{-1}$ при изменении элементов матрицы $\mathbf{A}$. Матрицу $\mathbf{A}^{-1}$ называют *устойчивой*, если малым изменениям элементов $\mathbf{A}$ отвечают малые изменения элементов $\mathbf{A}^{-1}$, и *неустойчивой* — в противном случае. Матрица называется *плохо обусловленной*, если ее обратная матрица неустойчива.

Этот факт прямо связан с "неприятностями" при решении системы (2.0.1), даже когда она имеет единственное решение ($\mathbf{x} = \mathbf{A}^{-1}\mathbf{b}$). Если матрица плохо обусловлена, то малое изменение элементов $\mathbf{A}$ или $\mathbf{b}$ приводит к заметному изменению решения. Получим количественную характеристику этого явления.

Первоначально будем считать, что матрица $\mathbf{A}$ известна точно, а вектор $\mathbf{b}$ — с некоторой погрешностью $\Delta\mathbf{b}$. Тогда система приобретет вид

$$\mathbf{A}(\mathbf{x} + \Delta\mathbf{x}) = \mathbf{b} + \Delta\mathbf{b}$$

или после вычитания (2.0.1) и обращения матрицы:

$$\mathbf{A}\Delta\mathbf{x} = \Delta\mathbf{b}; \quad \Delta\mathbf{x} = \mathbf{A}^{-1}\Delta\mathbf{b}.$$

Далее при использовании любой нормы матрицы, согласованной с нормой вектора $\mathbf{x}$, получаем

$$\|\Delta\mathbf{x}\| \le \|\mathbf{A}^{-1}\|\|\Delta\mathbf{b}\|, \quad \|\mathbf{b}\| \le \|\mathbf{A}\|\|\mathbf{x}\|.$$

Перемножение этих двух неравенств в предположении, что $\mathbf{b} \neq 0$, и деление на $\|\mathbf{b}\| \cdot \|\mathbf{x}\|$ дает

$$\frac{\|\Delta\mathbf{x}\|}{\|\mathbf{x}\|} \le \mathrm{cond}(\mathbf{A})\frac{\|\Delta\mathbf{b}\|}{\|\mathbf{b}\|}, \qquad \mathrm{cond}(\mathbf{A}) = \|\mathbf{A}\|\|\mathbf{A}^{-1}\|. \qquad (2.1.1)$$

Число $\mathrm{cond}(\mathbf{A}) = \|\mathbf{A}\|\|\mathbf{A}^{-1}\|$ будем называть *стандартным числом обусловленности.*

Вычисляя норму от обеих частей равенства $\mathbf{A} \cdot \mathbf{A}^{-1} = \mathbf{E}$, имеем $\|\mathbf{A}\| \cdot \|\mathbf{A}^{-1}\| \geq \mathbf{E}$, т. е. $\mathrm{cond}(\mathbf{A}) \geq 1$. Равенство (2.1.1) допускает простую интерпретацию для практики. Число обусловленности матрицы $\mathbf{A}$ является верхней границей "усиления" относительной ошибки вектора $\mathbf{b}$, т. е. относительное изменение вектора $\mathbf{b}$ влечет за собой относительное изменение в решении не более чем в $\mathrm{cond}(\mathbf{A})$ раз. Если величина $\mathrm{cond}(\mathbf{A})$ невелика, то говорят о хорошей обусловленности матрицы $\mathbf{A}$, в противном случае — о плохой.

На практике плохая обусловленность часто сопровождается малой величиной определителя матрицы $\det(\mathbf{A})$. В связи с этим широко распространено заблуждение, что малость $\det(\mathbf{A})$ всегда сопровождается большим числом обусловленности $\mathrm{cond}(\mathbf{A})$. Однако это не всегда так. Так, например, для матрицы $\mathbf{A} = \varepsilon\mathbf{E}$ ее определитель ($\det(\mathbf{A}) = \varepsilon^m$) может быть близок к нулю при больших размерах m матрицы $\mathbf{A}$ и сравнительно не малых $\varepsilon < 1$. Однако матрица $\mathbf{A}$ не является плохо обусловленной и не вызывает проблем с решением (2.0.1). Для норм $\|\bullet\|_1$ и $\|\bullet\|_\infty$ число обусловленности минимально $\left(\mathrm{cond}(\mathbf{A}) = \|\mathbf{A}\| \cdot \|\mathbf{A}^{-1}\| = 1\right)$, а для $\|\bullet\|_E$ оно не слишком велико: $\mathrm{cond}(\mathbf{A}) = \sqrt{m} \cdot \varepsilon \cdot \dfrac{\sqrt{m}}{\varepsilon} = m$.

Для иллюстрации рассмотрим систему (2.0.1) с параметрами

$$\mathbf{A} = \begin{pmatrix} 1.00 & 0.99 \\ 0.99 & 0.98 \end{pmatrix}, \ \mathbf{b} = \begin{pmatrix} 1.99 \\ 1.97 \end{pmatrix}, \qquad (2.1.2)$$

точным решением которой являются: $x^{(1)} = 1$, $x^{(2)} = 1$. Небольшое изменение в исходных данных резко изменяет решение:

$$\mathbf{b} + \Delta\mathbf{b} = \begin{pmatrix} 1.989903 \\ 1.970106 \end{pmatrix}, \ \Delta\mathbf{b} = \begin{pmatrix} -0.000097 \\ +0.000106 \end{pmatrix}, \ \Delta\mathbf{x} = \begin{pmatrix} +2.0000 \\ -2.0203 \end{pmatrix}, \ \mathbf{x} + \Delta\mathbf{x} = \begin{pmatrix} +3.0000 \\ -1.0203 \end{pmatrix}.$$

Непосредственное вычисление оценки из (2.1.1) дает $\mathrm{cond}(\mathbf{A}) \sim 40\,000$.

Для этого простого случая есть хорошая геометрическая иллюстрация. Каждому уравнению системы соответствует прямая на плоскости, а точка пересечения этих прямых дает решение системы. Исходным данным (2.1.2) каче-

ственно отвечает на рис. 2.1 случай (3): прямые пересекаются под очень острым углом, и малейшее изменение исходных коэффициентов **A** или **b** значительно влияет на расположение точки пересечения. Случай (1) демонстрирует хорошую обусловленность, а случай (2) — отсутствие решения при нулевом определителе.

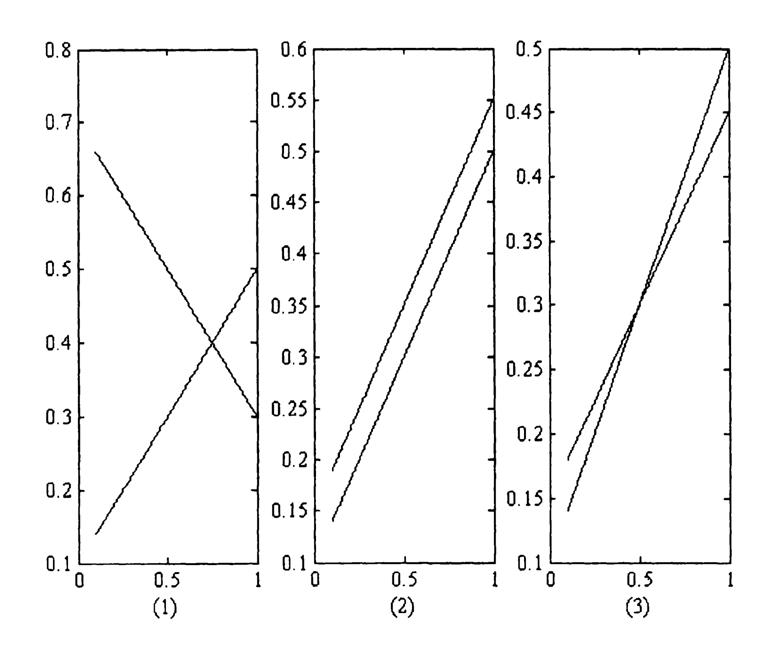


Рис. 2.1. Геометрическая иллюстрация решения линейной системы

Теперь рассмотрим ситуацию, когда вектор **b** известен точно, а коэффициенты матрицы **A** заданы с погрешностью $\Delta\mathbf{A}$:

$$(\mathbf{A}+\Delta\mathbf{A})(\mathbf{x}+\Delta\mathbf{x})=\mathbf{b}.$$

Вычитая из этой формулы равенство (2.0.1), получаем:

$$\mathbf{A}\Delta\mathbf{x}=-\Delta\mathbf{A}(\mathbf{x}+\Delta\mathbf{x}),$$

$$\|\Delta\mathbf{x}\|\le\|\mathbf{A}^{-1}\|\cdot\|\Delta\mathbf{A}\|\cdot\|\mathbf{x}+\Delta\mathbf{x}\|,$$

$$\frac{\|\Delta\mathbf{x}\|}{\|\mathbf{x}+\Delta\mathbf{x}\|}\le\mathrm{cond}(\mathbf{A})\frac{\|\Delta\mathbf{A}\|}{\|\mathbf{A}\|}. \tag{2.1.3}$$

И в этом случае $\text{cond}(\mathbf{A})$ ограничивает сверху увеличение относительной ошибки решения по сравнению с относительной ошибкой исходных данных. Если в примере (2.1.2) заменить элемент a_{22} на $a_{22} + \Delta a_{22} = 0.9802$, то "возмущенное" решение не будет совпадать с исходным даже по знаку ($x^{(1)} = 2.98$, $x^{(2)} = -1.00$).

В заключение следует отметить, что число обусловленности дает завышенную (иногда значительно) оценку погрешности. Трудности вычисления $\text{cond}(\mathbf{A})$ связаны с неприятностями при нахождении $\mathbf{A}^{-1}$. Если вычислять $\mathbf{A}^{-1}$, то время, требуемое для нахождения решения, например, методом Гаусса, заметно возрастает. Однако часто вместо $\text{cond}(\mathbf{A})$ вполне устраивает любая разумная оценка. Из многих применяемых способов оценки $\text{cond}(\mathbf{A})$ приведем один из широко используемых:

$$\text{cond}(\mathbf{A}) \approx \max_k \|\mathbf{a}_k\| \cdot \frac{\|\mathbf{z}\|}{\|\mathbf{y}\|},$$

где $\mathbf{y}$ и $\mathbf{z}$ — такие векторы, что $\|\mathbf{z}\|/\|\mathbf{y}\| \approx \mathbf{A}^{-1}$, $\mathbf{a}_k$ — столбцы матрицы $\mathbf{A}$. Векторы $\mathbf{y}$ и $\mathbf{z}$ находятся из решения двух систем: $\mathbf{A}^T\mathbf{y} = \mathbf{e}$ и $\mathbf{Az} = \mathbf{y}$, где $\mathbf{A}^T$ — транспонированная матрица, а $\mathbf{e}$ — вектор с компонентами, равными ± 1. Очевидны неравенства: $\|\mathbf{z}\| \le \|\mathbf{A}^{-1}\| \cdot \|\mathbf{y}\|$, $\|\mathbf{A}^{-1}\| \le \|\mathbf{z}\|/\|\mathbf{y}\|$. Вектор $\mathbf{e}$ выбирается специальным образом, чтобы увеличить надежность оценки.

Механизм влияния погрешности в задании элементов $\mathbf{A}$ на решение (2.0.1) интересно рассмотреть и с другой стороны, анализируя собственные значения $\mathbf{A}$ и $\mathbf{A}^{-1}$. Предварительно обратимся к двум положениям.

Положение 1. Максимальные по модулю элементы матрицы $\mathbf{A}$ имеют величину, по меньшей мере, порядка максимальных по модулю собственных значений матрицы $|\lambda_k|_{\max}$ (а может быть, и значительно их превышают). Действительно, это положение прямо следует из неравенства, связывающего собственные значения и норму матрицы: $|\lambda_k| \le \|\mathbf{A}\|$.

Положение 2. Пусть λ_k — собственные значения матрицы $\mathbf{A}$. В "возмущенной" матрице $\mathbf{A} + \Delta\mathbf{A}$ собственные значения могут измениться на величину порядка элементов матрицы возмущений $\Delta\mathbf{A}$.

В качестве примера достаточно обратиться к матрице $\mathbf{A}+\Delta\mathbf{A}=\mathbf{A}+\varepsilon\mathbf{E}$, все собственные значения которой изменились на одну величину: $\Delta\lambda_k=\varepsilon$.

На практике положение 2 часто имеет место, хотя можно специально таким образом выполнить достаточно большое возмущение, что в матрицах $\mathbf{A}$ и $\mathbf{A}+\Delta\mathbf{A}$ собственные значения не изменятся. Например:

$$\mathbf{A}=\begin{pmatrix}1 & -1\\ 2 & 4\end{pmatrix},\quad \mathbf{A}+\Delta\mathbf{A}=\begin{pmatrix}5 & 1\\ -6 & 0\end{pmatrix},\quad \lambda_1=2,\quad \lambda_2=3.$$

Пусть теперь предельная относительная погрешность задания элементов матрицы $\mathbf{A}$ равна δ. Тогда в соответствии с положением 1 максимальные по модулю элементы матрицы $\Delta\mathbf{A}$, по меньшей мере, имеют величину порядка $\delta|\lambda_k|_{\max}$. В соответствии с положением 2 на эту величину могут измениться все собственные значения возмущенной матрицы.

Если величина δ мала, то максимальные по модулю собственные значения матрицы $\mathbf{A}$ изменятся незначительно. В то же время, если разброс между λ_k велик, минимальные по модулю собственные значения λ_k могут измениться весьма существенно. Но величина, обратная минимальному по модулю собственному значению матрицы $\mathbf{A}$, является максимальным по модулю собственным значением для обратной матрицы $\mathbf{A}^{-1}$, элементы которой претерпят существенные изменения. Вместе с ними изменятся весьма сильно и компоненты вектора решения системы (2.0.1).

Описанный эффект будет проявляться тем сильнее, чем больше разброс собственных значений матрицы $\mathbf{A}$, что в свою очередь, как и $\mathrm{cond}(\mathbf{A})$, может служить количественной характеристикой плохой обусловленности матрицы.

Поскольку $\max\limits_k|\lambda_k|\le\|\mathbf{A}\|$, а также $\max\limits_k\dfrac{1}{|\lambda_k|}\le\|\mathbf{A}^{-1}\|$, для плохо обусловленных матриц имеем:

$$\mathrm{cond}(\mathbf{A})=\|\mathbf{A}\|\|\mathbf{A}^{-1}\|\ge\max_k|\lambda_k|\cdot\max_k|\lambda_k|^{-1}=\frac{\max\limits_k|\lambda_k|}{\min\limits_k|\lambda_k|}\gg 1.$$

Теперь попытаемся ввести число обусловленности несколько иначе. Пусть возмущенная задача имеет вид:

$$\mathbf{A}(\mathbf{x}+\Delta\mathbf{x})=\mathbf{b}+\mathbf{r}, \tag{2.1.4}$$

где $\mathbf{r}$ отражает оба возмущения (и в матрице $\mathbf{A}$, и в векторе $\mathbf{b}$), т. е. возмущение в матрице $\Delta\mathbf{A}$ перенесено в правую часть и включено в $\mathbf{r}$. Это не совсем правильно, т. к. $\Delta\mathbf{A}$ умножается на $\mathbf{x}$, но мы пытаемся получить лишь грубую оценку. Вычитая (2.0.1) из (2.0.4), имеем

$$\mathbf{A}\Delta\mathbf{x} = \mathbf{r}$$

или

$$\Delta\mathbf{x} = \mathbf{A}^{-1}\mathbf{r}.$$

Перейдя к нормам, получим

$$\|\Delta\mathbf{x}\| = \left\|\mathbf{A}^{-1}\mathbf{r}\right\| \le \left\|\mathbf{A}^{-1}\right\| \cdot \|\mathbf{r}\|.$$

Это оценка абсолютной погрешности. Для относительной погрешности она имеет вид

$$\frac{\|\Delta\mathbf{x}\|}{\|\mathbf{x}\|} \le \frac{\left\|\mathbf{A}^{-1}\right\| \cdot \|\mathbf{r}\| \cdot \|\mathbf{b}\|}{\|\mathbf{x}\| \cdot \|\mathbf{b}\|} = \mathrm{cond}_e(\mathbf{A})\frac{\|\mathbf{r}\|}{\|\mathbf{b}\|}, \text{ где } \mathrm{cond}_e(\mathbf{A}) = \frac{\left\|\mathbf{A}^{-1}\right\| \cdot \|\mathbf{b}\|}{\|\mathbf{x}\|}. \quad (2.1.5)$$

Это неравенство определяет *естественное* число обусловленности. Оно не очень удобно, т. к. является апостериорным (оно зависит от $\|\mathbf{x}\|$, а $\mathbf{x}$ не известен априори). Исключение $\mathbf{x}$ приведет к *стандартному* числу обусловленности. Это делается следующим образом. Из (2.0.1) следует

$$\|\mathbf{b}\| \le \|\mathbf{A}\| \cdot \|\mathbf{x}\| \quad \text{или} \quad \frac{1}{\|\mathbf{x}\|} \le \frac{\|\mathbf{A}\|}{\|\mathbf{b}\|}.$$

Подставляя это неравенство в (2.1.5), получаем уже известную оценку

$$\frac{\|\Delta\mathbf{x}\|}{\|\mathbf{x}\|} \le \frac{\left\|\mathbf{A}^{-1}\right\| \cdot \|\mathbf{b}\|}{\|\mathbf{x}\|} \cdot \frac{\|\mathbf{r}\|}{\|\mathbf{b}\|} \le \frac{\left\|\mathbf{A}^{-1}\right\| \cdot \|\mathbf{b}\| \cdot \|\mathbf{A}\|}{\|\mathbf{b}\|} \cdot \frac{\|\mathbf{r}\|}{\|\mathbf{b}\|} = \|\mathbf{A}\| \cdot \left\|\mathbf{A}^{-1}\right\| \cdot \frac{\|\mathbf{r}\|}{\|\mathbf{b}\|}.$$

Обратим внимание еще на один аспект интерпретации неравенств (2.1.1) и (2.1.3). Если значение $\mathrm{cond}(\mathbf{A})$ мало (близко к единице), то относительно малое изменение исходных данных *обязательно* приведет лишь к малому изменению решения. Если же значение $\mathrm{cond}(\mathbf{A})$ велико, то малые изменения исходных данных *могут* привести (но не обязательно приводят) к большому изменению решения.

В зависимости от числа двоичных разрядов, представляющих вещественные числа в данном компьютере, существует "критическое" число обусловленности. Если, например, $\mathrm{cond}(\mathbf{A}) = 10^6$, то в решении может быть потеряно

шесть десятичных знаков. При длине мантиссы в семь десятичных разрядов это оказывается "вычислительной катастрофой", в то время как при работе с двойной точностью (15 десятичных разрядов) проблем может не быть. Формализовать это можно следующим образом.

Точность машинной арифметики характеризуется посредством "машинного эпсилон", т. е. наименьшего числа с плавающей точкой ε, такого, что $1 \oplus \varepsilon > 1$, где $\oplus$ — сложение на компьютере. При записи в оперативную память элементов матрицы $\mathbf{A}$ предельная относительная погрешность составляет не менее ε. Как следует из приведенных ранее утверждений, только этого искажения исходной информации вполне достаточно, чтобы решение не имело ни одного верного знака, если число обусловленности удовлетворяет условию $\mathrm{cond}(\mathbf{A}) > 1/\varepsilon$ (разумеется, здесь предполагается, что $\mathrm{cond}(\mathbf{A})$ не является слишком завышенной мерой плохой обусловленности). Последнее замечание не излишне, поскольку при решении системы (2.0.1) с диагональной матрицей $\mathbf{A}$ не возникает проблем, в то время как $\mathrm{cond}(\mathbf{A})$ может быть очень большим:

$$\mathbf{A} = \begin{pmatrix} 10^4 & 0 \\ 0 & 10^{-4} \end{pmatrix}, \quad \mathbf{A}^{-1} = \begin{pmatrix} 10^{-4} & 0 \\ 0 & 10^4 \end{pmatrix}, \quad \mathrm{cond}(\mathbf{A}) = 10^8.$$

2.2. Метод Гаусса. LU-разложение матрицы. Программы *DECOMP* и *SOLVE*

Различают два больших класса методов решения систем (2.0.1): *точные* (или *прямые*) и *итерационные*. Точные методы за конечное число арифметических операций при отсутствии ошибок округления (что эквивалентно бесконечной разрядной сетке) дают точное решение задачи. В ходе применения итерационных методов рождается последовательность векторов, сходящаяся к решению.

В качестве наиболее популярного представителя методов первой группы рассмотрим метод Гаусса исключения неизвестных. Одна из его примитивных модификаций предполагает на первом шаге исключение $x^{(1)}$ с помощью первого уравнения из остальных уравнений. С этой целью первое уравнение умножается на $m_{k1} = -a_{k1}/a_{11}$ и складывается с k-м уравнением и т. д. На втором шаге с помощью преобразованного второго уравнения исключается $x^{(2)}$ из последующих уравнений. После исключения $x^{(n-1)}$ завершается так назы-

ваемый *прямой ход* метода Гаусса, результатом которого является верхняя треугольная матрица. *Обратный ход* метода Гаусса (гораздо менее трудоемкий) сводится к последовательному получению неизвестных, начиная с последнего уравнения.

Алгоритм в таком виде нуждается в существенном замечании. Нельзя заранее предвидеть, что элемент, стоящий в левом верхнем углу обрабатываемой матрицы, всегда будет отличен от нуля. Если ситуация с нулевым элементом возникнет, то, чтобы избежать деления на ноль, необходимо переставить строки, сделав элемент в этой позиции (ведущий элемент) ненулевым. Более того, желательно избегать не только нулевых, но и относительно малых ведущих элементов. Подтвердим это примером:

$$0.100 \cdot 10^{-3} \cdot x^{(1)} + 0.100 \cdot 10^{1} \cdot x^{(2)} = 0.100 \cdot 10^{1},$$

$$0.100 \cdot 10^{1} \cdot x^{(1)} + 0.100 \cdot 10^{1} \cdot x^{(2)} = 0.200 \cdot 10^{1}.$$

Решение с малым ведущим элементом $a_{11} = 0.100 \cdot 10^{-3}$ с шестью десятичными знаками таково: $x^{(1)} = 1.00010$, $x^{(2)} = 0.999900$, а с тремя знаками: $x^{(1)} = 0.000$, $x^{(2)} = 1.00$. Очевидно, что произошла "вычислительная катастрофа". Переставив уравнения (теперь $a_{11} = 0.100 \cdot 10^{1}$), решим систему с тремя значащими цифрами: $x^{(1)} = 1.00$, $x^{(2)} = 1.00$. (Рекомендуется самостоятельно проделать вычисления, корректно округляя.)

Наиболее известны следующие две стратегии выбора ведущего элемента.

- Вариант 1 — *полный* выбор. Здесь на k-ом шаге в качестве ведущего берется наибольший по модулю элемент в неприведенной части матрицы. Затем строки и столбцы переставляются так, чтобы этот элемент поменялся местами с a_{kk}. В этом случае каждый раз осуществляется деление на максимальный элемент, но перестановка столбцов фактически сводится к перенумерации компонентов вектора $\mathbf{x}$.
- Вариант 2 — *частичный* выбор. Здесь на k-ом шаге в качестве ведущего используют наибольший по модулю элемент первого столбца неприведенной части. Затем этот элемент меняют местами с a_{kk}, для чего переставляют только строки, избегая перенумерации компонентов вектора $\mathbf{x}$.

В ходе многочисленных машинных экспериментов установлено, что, как правило, частичный выбор лишь немного уступает полному в скорости роста ошибок округления. При этом полный выбор гораздо более трудоемок, тре-

бует перенумерации переменных при перестановке столбцов и увеличивает время получения решения.

С современной точки зрения метод Гаусса интерпретируется как разложение матрицы системы (2.0.1) в произведение двух треугольных матриц (LU-разложение). Этот факт отражает следующая теорема, приводимая без доказательства.

Теорема. Пусть $\mathbf{A}^{(k)}$ — главные миноры квадратной матрицы $\mathbf{A}$ порядка $m \times m$ ($k = 1, 2, ..., m-1$). Предположим, что $\det\left(\mathbf{A}^{(k)}\right) \neq 0$. Тогда существуют единственная нижняя треугольная матрица $\mathbf{L} = (l_{ij})$, где $l_{11} = l_{22} = ... = l_{nn} = 1$, и единственная верхняя треугольная матрица $\mathbf{U} = (u_{ij})$, такие, что $\mathbf{LU} = \mathbf{A}$. Более того, $\det(\mathbf{A}) = u_{11} \cdot u_{22} \cdot ... \cdot u_{nn}$.

Эта теорема позволяет представить решение (2.0.1) как решение двух систем с треугольными матрицами $\mathbf{L}$ и $\mathbf{U}$: $\mathbf{Ly} = \mathbf{b}$ и $\mathbf{Ux} = \mathbf{y}$. Решение первой системы с одновременным вычислением $\mathbf{L}$ и $\mathbf{U}$ соответствует прямому ходу метода Гаусса, а решение второй системы — обратному ходу. Технологию LU-разложения проиллюстрируем на примере системы четвертого порядка без выбора ведущего элемента. Пусть $m_{k1} = -a_{k1}/a_{11}$ ($k = 2, 3, 4$). Первый шаг прямого хода эквивалентен умножению матрицы $\mathbf{A}$ и вектора $\mathbf{b}$ слева на матрицу $\mathbf{M}_1$:

$$\mathbf{M}_1 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ m_{21} & 1 & 0 & 0 \\ m_{31} & 0 & 1 & 0 \\ m_{41} & 0 & 0 & 1 \end{pmatrix}, \quad \mathbf{A}_2 = \mathbf{M}_1\mathbf{A}, \quad \mathbf{b}_2 = \mathbf{M}_1\mathbf{b}.$$

На втором шаге матрица $\mathbf{A}_2$ и вектор $\mathbf{b}_2$ умножаются на матрицу $\mathbf{M}_2$, а на третьем шаге матрица $\mathbf{A}_3 = \mathbf{M}_2\mathbf{A}_2$ и вектор $\mathbf{b}_3 = \mathbf{M}_2\mathbf{b}_2$ умножаются на матрицу $\mathbf{M}_3$ ($\mathbf{A}_4 = \mathbf{M}_3\mathbf{M}_2\mathbf{M}_1\mathbf{A}$): $m_{k2} = -a_{k2}^{(2)}/a_{22}^{(2)}$, $m_{k3} = -a_{k3}^{(2)}/a_{33}^{(2)}$,

$$\mathbf{M}_2 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & m_{32} & 1 & 0 \\ 0 & m_{33} & 0 & 1 \end{pmatrix}, \quad \mathbf{M}_3 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & m_{43} & 1 \end{pmatrix}.$$

Согласно построению $\mathbf{A}_4$ есть верхняя треугольная матрица $\mathbf{U}$:

$$\mathbf{M} = \mathbf{M}_3\mathbf{M}_2\mathbf{M}_1, \quad \mathbf{MA} = \mathbf{U}, \quad \mathbf{L} = \mathbf{M}^{-1}, \quad \mathbf{A} = \mathbf{LU},$$

где

$$\mathbf{L} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ -m_{21} & 1 & 0 & 0 \\ -m_{31} & -m_{32} & 1 & 0 \\ -m_{41} & -m_{42} & -m_{43} & 1 \end{pmatrix}.$$

Здесь учитывалось, что произведение однотипных треугольных матриц — это треугольная матрица того же типа, а обратная матрица для треугольной (нижней или верхней) является треугольной того же типа.

Теперь сформулируем ряд условий, которым должна удовлетворять современная программа, реализующая метод Гаусса. Нужно предусмотреть: выбор ведущего элемента, эффективное решение нескольких систем уравнений (2.0.1) с одной и той же матрицей $\mathbf{A}$ и различными векторами $\mathbf{b}$, оценку числа обусловленности. Реализованные в большинстве пакетов по линейной алгебре программы представляют собой набор из двух программ. В первой осуществляется LU-разложение, а во второй решаются две системы с треугольными матрицами $\mathbf{L}$ и $\mathbf{U}$ ($\mathbf{Ly} = \mathbf{b}$ и $\mathbf{Ux} = \mathbf{y}$). Примером являются написанные на Фортране программы DECOMP и SOLVE из [14]. Программы имеют следующие параметры:

```
DECOMP (NDIM, N, A, COND, IPVT, WORK)
SOLVE(NDIM, N, A, B, IPVT)
```

где:

- NDIM — объявленная в описании строчная размерность массива, в котором располагается матрица $\mathbf{A}$;
- N — порядок системы уравнений;
- A — матрица, подвергающаяся разложению (по окончании работы программы на ее месте располагаются матрицы $\mathbf{L}$ и $\mathbf{U}$),
- COND — оценка числа обусловленности;
- IPVT — вектор индексов ведущих элементов (размерность N);
- WORK — рабочий одномерный массив (размерность N),
- B — вектор правых частей системы (2.0.1), где по окончании работы программы SOLVE размещается вектор решения $\mathbf{x}$.

Иногда функции DECOMP и SOLVE возлагаются на одну программу, имеющую в таком случае управляющий параметр. В ряде случаев предлагается совокупность разнообразных программ для решения систем (2.0.1) с различными матрицами (общего вида, ленточными, трехдиагональными, положительно определенными, симметрическими и пр.).

В заключение оценим число арифметических операций в методе Гаусса. На каждом шаге исключения мы встречаемся с операциями деления и умножения-вычитания. Возьмем за единицу измерения операцию именно такого типа. На k-ом шаге в одной строке выполняется одно деление и k умножений-вычитаний. Тогда для всех $k-1$ строк имеем: $(k+1)(k-1)=k^2-1$ операций. В прямом ходе Гаусса таких шагов m. В итоге получаем:

$$\sum_{k=1}^{m}\left(k^2-1\right)=\sum_{k=1}^{m}k^2-m=\frac{2m^3+3m^2-5m}{6}.$$

При больших значениях m хорошим приближением для числа операций будет $m^3/3$. Для обратного хода нужно на порядок меньше операций (одно деление и $k-1$ умножение-вычитание при вычислении $x^{(k)}$, что для всех компонентов дает величину $\sum_{k=1}^{m}k=\frac{m^2+m}{2}$). Для сравнения: в формуле Крамера требуется выполнить $m!\left(m^2-1\right)$ операций. (Вычисление одного определителя m-ого порядка требует $m!(m-1)$ умножений, а всего нужно вычислить $m+1$ определитель, значит, всего $m!\left(m^2-1\right)$. Если $m=10$, то для LU-разложения число операций $10^3/3$, а для формул Крамера $10!\times 99\cong 3\times 10^8$.)

2.3. Итерационные методы

Итерационные методы (еще одно название — *методы последовательных приближений*) дают возможность для системы (2.0.1) $\mathbf{Ax}=\mathbf{b}$ строить последовательность векторов $\mathbf{x}_0$, $\mathbf{x}_1$, ..., $\mathbf{x}_n$, ... пределом которой должно быть точное решение $\mathbf{x}^*$.

$$\mathbf{x}^*=\lim_{n\to\infty}\mathbf{x}_n. \qquad (2.3.1)$$

На практике построение последовательности обрывается, как только достигается желаемая точность. Чаще всего для достаточно малого значения $\varepsilon > 0$ контролируется выполнение оценки $\left|\mathbf{x}^* - \mathbf{x}_n\right| < \varepsilon$. Метод последовательных приближений может быть построен, например, по следующей схеме. Эквивалентными преобразованиями приведем систему (2.0.1) к виду

$$\mathbf{x} = \mathbf{C}\mathbf{x} + \mathbf{d}. \qquad (2.3.2)$$

Под эквивалентными преобразованиями будем понимать преобразования, сохраняющие решение системы (т. е. решения (2.0.1) и (2.3.2) совпадают). Точное решение $\mathbf{x}^*$ системы (2.3.2) имеет вид

$$\mathbf{x}^* = \left(\mathbf{E} - \mathbf{C}\right)^{-1}\mathbf{d}. \qquad (2.3.3)$$

Вместо (2.3.2) будем решать систему разностных уравнений (2.3.4)

$$\mathbf{x}_{n+1} = \mathbf{C}\mathbf{x}_n + \mathbf{d} \qquad (2.3.4)$$

пошаговым методом. При этом необходимо решить целый ряд вопросов. Сходится ли итерационный процесс (2.3.4)? Если сходится, что является пределом последовательности и какова скорость сходимости?

Легко заметить, что система (2.3.4) имеет вид (П3.18). Поэтому в соответствии с (П3.20) решение (2.3.4) записывается в виде

$$\mathbf{x}_n = \mathbf{C}^n\mathbf{x}_0 + \left(\mathbf{E} - \mathbf{C}^n\right)\left(\mathbf{E} - \mathbf{C}\right)^{-1}\mathbf{d}. \qquad (2.3.5)$$

Вычитая из (2.3.5) точное решение (2.3.3), получаем

$$\mathbf{x}_n - \mathbf{x}^* = \mathbf{C}^n\mathbf{x}_0 - \mathbf{C}^n\left(\mathbf{E} - \mathbf{C}\right)^{-1}\mathbf{d} = \mathbf{C}^n\left(\mathbf{x}_0 - \mathbf{x}^*\right). \qquad (2.3.6)$$

Чтобы обеспечить условие сходимости (2.3.1), все элементы матрицы $\mathbf{C}^n$ должны стремиться к нулю при $n \to \infty$. Для этого, в свою очередь, необходимо и достаточно, чтобы *все* собственные значения матрицы **C** были бы по модулю меньше единицы:

$$\left|\lambda_k\right| < 1. \qquad (2.3.7)$$

Последний факт подробно поясняется в *разд. П3.9*, где определяются условия асимптотической устойчивости решения системы (2.3.4). Поскольку нахождение всех собственных значений доставляет значительные трудности, с учетом (П3.9) вместо условия (2.3.7) можно использовать достаточное условие сходимости

$$\left\|\mathbf{C}\right\| < 1, \qquad (2.3.8)$$

которое справедливо для любой канонической нормы.

Количество итераций по формуле (2.3.4) будет тем меньше, чем меньше по модулю собственные значения матрицы $\mathbf{C}$ и чем ближе к $\mathbf{x}^*$ выбрано начальное приближение $\mathbf{x}_0$. На практике при реализации на компьютере процесс (2.3.4) прерывается либо заданием максимального числа итераций, либо условием $\|\mathbf{x}_{n+1} - \mathbf{x}_n\| < \varepsilon$. Таким образом, основным неформальным моментом является такое приведение системы (2.0.1) к виду (2.3.2), чтобы выполнялось условие (2.3.8). В общем случае универсальный способ такого перехода с малой трудоемкостью отсутствует, и поэтому часто используется специфика решаемой задачи. Рассмотрим следующий пример.

Пусть диагональные элементы матрицы $\mathbf{A}$ в (2.0.1) значительно превышают по модулю остальные элементы в соответствующих строках. Разделим каждое уравнение на соответствующий диагональный элемент и получим

$$\tilde{\mathbf{A}}\mathbf{x} = \tilde{\mathbf{b}}, \quad \mathbf{x} = \left(\mathbf{E} - \tilde{\mathbf{A}}\right)\mathbf{x} + \tilde{\mathbf{b}}.$$

На главной диагонали у матрицы $\tilde{\mathbf{A}}$ стоят единицы, а у матрицы $\left(\mathbf{E} - \tilde{\mathbf{A}}\right)$ расположены нули. Вне главной диагонали у обеих матриц находятся малые по модулю элементы, что позволяет, выбрав $\mathbf{C} = \mathbf{E} - \tilde{\mathbf{A}}$, легко обеспечить условие (2.3.8) и быструю сходимость итерационного процесса (2.3.4).

Рассмотрим несколько примеров итерационных методов. С этой целью, предполагая, что все диагональные элементы матрицы $\mathbf{A}$ отличны от нуля, преобразуем (2.0.1) к виду

$$x^{(k)} = -\sum_{j=1}^{k-1} \frac{a_{kj}}{a_{kk}} x^{(j)} - \sum_{j=k+1}^{m} \frac{a_{kj}}{a_{kk}} x^{(j)} + \frac{b^{(k)}}{a_{kk}}, \quad k = 1, 2, \ldots, m. \tag{2.3.9}$$

Здесь матрица $\mathbf{A}$ имеет размер $m \times m$, а $x^{(k)}$ — k-ый компонент вектора $\mathbf{x}$. Для $x^{(1)}$ первая сумма в (2.3.9) отсутствует. Как и ранее, обозначая номер итерации нижним индексом, вместо алгебраической системы (2.3.9) пошаговым методом будем решать следующую систему разностных уравнений:

$$x_{n+1}^{(k)} = -\sum_{j=1}^{k-1} \frac{a_{kj}}{a_{kk}} x_n^{(j)} - \sum_{j=k+1}^{m} \frac{a_{kj}}{a_{kk}} x_n^{(j)} + \frac{b^{(k)}}{a_{kk}}, \quad k = 1, 2, \ldots, m. \tag{2.3.10}$$

Этот метод получил название *метода Якоби*. Анализируя формулу (2.3.10), легко заметить, что к моменту вычисления $x_{n+1}^{(k)}$ значения $x_{n+1}^{(j)}$ для $j < k$ уже определены.

Это позволяет модифицировать (2.3.10)

$$x_{n+1}^{(k)} = -\sum_{j=1}^{k-1} \frac{a_{kj}}{a_{kk}} x_{n+1}^{(j)} - \sum_{j=k+1}^{m} \frac{a_{kj}}{a_{kk}} x_{n}^{(j)} + \frac{b^{(k)}}{a_{kk}}, \ k = 1, 2, ..., m \qquad (2.3.11)$$

и получить *метод Гаусса — Зейделя.*

Оба эти метода можно записать в матричном виде, вводя следующие обозначения. Пусть $\mathbf{D}$ — диагональная, $\mathbf{A}_1$ — левая треугольная, а $\mathbf{A}_2$ — правая треугольная матрицы. $\mathbf{A}_1$ и $\mathbf{A}_2$ имеют нулевую главную диагональ. Ненулевые элементы всех трех матриц совпадают с соответствующими элементами матрицы $\mathbf{A}$, и можно записать $\mathbf{A} = \mathbf{A}_1 + \mathbf{D} + \mathbf{A}_2$. Тогда формула (2.3.10) записывается следующим образом:

$$\mathbf{x}_{n+1} = -\mathbf{D}^{-1}\mathbf{A}_1\mathbf{x}_n - \mathbf{D}^{-1}\mathbf{A}_2\mathbf{x}_n + \mathbf{D}^{-1}\mathbf{b}$$

или

$$\mathbf{D}\left(\mathbf{x}_{n+1} - \mathbf{x}_n\right) + \mathbf{A}\mathbf{x}_n = \mathbf{b}, \qquad (2.3.12)$$

а (2.3.11) принимает вид:

$$\mathbf{x}_{n+1} = -\mathbf{D}^{-1}\mathbf{A}_1\mathbf{x}_{n+1} - \mathbf{D}^{-1}\mathbf{A}_2\mathbf{x}_n + \mathbf{D}^{-1}\mathbf{b}$$

или

$$\left(\mathbf{D} + \mathbf{A}_1\right)\left(\mathbf{x}_{n+1} - \mathbf{x}_n\right) + \mathbf{A}\mathbf{x}_n = \mathbf{b}. \qquad (2.3.13)$$

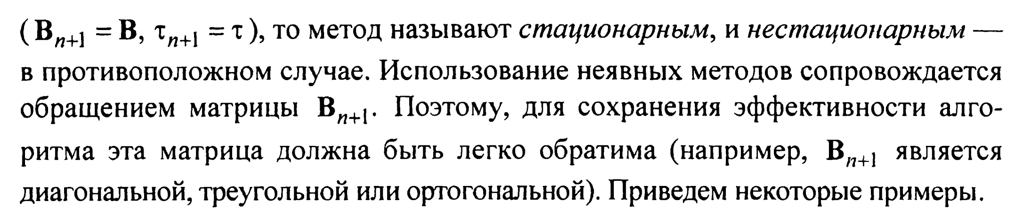
Вид формул (2.3.12) и (2.3.13) подсказывает путь к возможному обобщению. С этой целью добавим к левой части (2.0.1) следующее нулевое слагаемое:

$$\mathbf{B}\frac{\mathbf{x} - \mathbf{x}}{\tau} + \mathbf{A}\mathbf{x} = \mathbf{b}. \qquad (2.3.14)$$

Теперь вместо системы алгебраических уравнений (2.3.14) предлагается решать следующую систему разностных уравнений пошаговым методом:

$$\mathbf{B}_{n+1}\frac{\mathbf{x}_{n+1} - \mathbf{x}_n}{\tau_{n+1}} + \mathbf{A}\mathbf{x}_n = \mathbf{b}. \qquad (2.3.15)$$

Формула (2.3.15) называется *канонической формой одношагового итерационного метода* и охватывает алгоритмы, чьи разностные уравнения в матричной форме имеют первый порядок. Здесь $\mathbf{B}_{n+1}$ — матрица, задающая итерационный метод, τ_{n+1} — итерационный параметр. Если $\mathbf{B}_{n+1} = \mathbf{E}$, то метод называют *явным*, а в противном случае — *неявным*. Если матрица $\mathbf{B}_{n+1}$ и итерационный параметр τ_{n+1} постоянные и не зависят от номера итерации



($\mathbf{B}_{n+1} = \mathbf{B}$, $\tau_{n+1} = \tau$), то метод называют *стационарным*, и *нестационарным* — в противоположном случае. Использование неявных методов сопровождается обращением матрицы $\mathbf{B}_{n+1}$. Поэтому, для сохранения эффективности алгоритма эта матрица должна быть легко обратима (например, $\mathbf{B}_{n+1}$ является диагональной, треугольной или ортогональной). Приведем некоторые примеры.

- *Метод простой итерации.* Здесь $\mathbf{B}_{n+1} = \mathbf{E}$, а $\tau_{n+1} = \tau$.
- *Итерационный метод Ричардсона.* Здесь $\mathbf{B}_{n+1} = \mathbf{E}$, а τ_{n+1} — переменный параметр.
- *Метод Якоби.* $\mathbf{B}_{n+1} = \mathbf{D}$, а $\tau_{n+1} = 1$.
- *Метод верхней релаксации.* $\mathbf{B}_{n+1} = \mathbf{D} + \omega\mathbf{A}_1$, $\tau_{n+1} = \omega$, $0 < \omega < 2$, где ω — заданный числовой параметр. Для $\omega = 1$, как частный случай, получается *метод Гаусса — Зейделя.* Для симметрических положительно определенных матриц $\mathbf{A}$ условие $0 < \omega < 2$ является условием сходимости метода.

Скорость сходимости явных итерационных методов часто оставляет желать лучшего. Если матрица $\mathbf{A}$ плохо обусловлена, то требуемое число итераций целого ряда методов прямо пропорционально числу обусловленности. Для повышения скорости сходимости используются два пути. Первый из них предполагает применение нестационарных методов с эффективным выбором параметра τ_{n+1} на каждом шаге. Второй путь — использование неявных методов. Формально анализ свойств неявного метода может быть сведен к рассмотрению явного метода с матрицей $\mathbf{B}^{-1}\mathbf{A}$. Эффективный выбор $\mathbf{B}$ позволяет заметно уменьшить число обусловленности. Возможна и комбинация обоих подходов. Подробное освещение этих и других вопросов, касающихся свойств итерационных методов, можно найти в [13].

Остановимся на сравнении прямых и итерационных методов. Чаще всего преимущества итерационных методов сказываются в следующих ситуациях.

- Имеется хорошее начальное приближение $\mathbf{x}_0$ к точному решению (2.0.1), что обеспечит сравнительно малое число итераций в (2.3.15).
- Удалось получить матрицу $\mathbf{C}$ в (2.3.4) с весьма малыми по модулю собственными значениями, что гарантирует высокую скорость сходимости итерационного процесса.
- Матрица $\mathbf{A}$ является разреженной, и в оперативной памяти компьютера хранится относительно небольшое число ее ненулевых элементов. Ис-

пользование формулы (2.3.15) требует лишь написания специальной программы умножения разреженной матрицы на вектор, в то время как в процессе реализации точных методов и преобразования матрицы $\mathbf{A}$ может происходить заметное увеличение ненулевых элементов.

В заключение данного раздела рассмотрим популярного представителя так называемых методов вариационного типа — *метод минимальных невязок*. Формула этого метода полностью отвечает (2.3.15). Здесь вводится вектор невязки $\mathbf{r}_n = \mathbf{A}\mathbf{x}_n - \mathbf{b}$, характеризующий удаление $\mathbf{x}_n$ от точного решения системы (2.0.1). Каждое новое приближение вычисляется по формуле

$$\mathbf{x}_{n+1} = \mathbf{x}_n - \tau_{n+1}\mathbf{r}_n, \qquad (2.3.16)$$

а параметр τ_{n+1} выбирается из условия минимума длины вектора невязки на каждом шаге. Умножим (2.3.16) на матрицу $\mathbf{A}$ и вычтем из обеих частей равенства вектор $\mathbf{b}$:

$$\mathbf{r}_{n+1} = \mathbf{r}_n - \tau_{n+1}\mathbf{A}\mathbf{r}_n.$$

Тогда для скалярного произведения $(\mathbf{r}_{n+1}, \mathbf{r}_{n+1})$ имеем

$$(\mathbf{r}_{n+1}, \mathbf{r}_{n+1}) = (\mathbf{r}_n - \tau_{n+1}\mathbf{A}\mathbf{r}_n, \mathbf{r}_n - \tau_{n+1}\mathbf{A}\mathbf{r}_n) =$$
$$= (\mathbf{r}_n, \mathbf{r}_n) - 2\tau_{n+1}(\mathbf{A}\mathbf{r}_n, \mathbf{r}_n) + \tau_{n+1}^2(\mathbf{A}\mathbf{r}_n, \mathbf{A}\mathbf{r}_n).$$

Удовлетворяя условие минимума и дифференцируя это выражение по τ_{n+1} с приравниванием нулю производной, получаем

$$-2(\mathbf{A}\mathbf{r}_n, \mathbf{r}_n) + 2\tau_{n+1}(\mathbf{A}\mathbf{r}_n, \mathbf{A}\mathbf{r}_n) = 0,$$

$$\tau_{n+1} = \frac{(\mathbf{A}\mathbf{r}_n, \mathbf{r}_n)}{(\mathbf{A}\mathbf{r}_n, \mathbf{A}\mathbf{r}_n)}. \qquad (2.3.17)$$

Выражения (2.3.16) и (2.3.17) являются рабочими формулами метода.

Методы (2.3.15) являются одношаговыми. Повышение порядка разностного уравнения позволяет получить качественно иные результаты.

2.4. Метод сопряженных градиентов

Метод сопряженных градиентов предназначен для решения системы (2.0.1) с симметрической положительно определенной матрицей $\mathbf{A}$. Для изложения

его рабочих формул необходимо ввести понятие *A-ортогональности*. Последовательность векторов $\{\mathbf{s}_k\}$ называется A-ортогональной, если $(\mathbf{A}\mathbf{s}_k, \mathbf{s}_j) = 0$ для $k \neq j$.

Суть метода сводится к следующему. На основе последовательно вычисляемых линейно независимых векторов невязок $\mathbf{r}_0$, $\mathbf{r}_1$, ..., $\mathbf{r}_n$, ...

$$\mathbf{r}_n = \mathbf{A}\mathbf{x}_n - \mathbf{b} \tag{2.4.1}$$

одновременно с их получением с использованием процедуры Грама — Шмидта, строится система A-ортогональных векторов $\mathbf{s}_k$:

$$\mathbf{s}_1 = \mathbf{r}_0, \quad \mathbf{s}_{n+1} = \mathbf{r}_n - \sum_{j=1}^{n} b_{nj}\mathbf{s}_j; \quad b_{nj} = \frac{(\mathbf{A}\mathbf{s}_j, \mathbf{r}_n)}{(\mathbf{A}\mathbf{s}_j, \mathbf{s}_j)}. \tag{2.4.2}$$

В этих обозначениях итерационный метод записывается в виде:

$$\mathbf{x}_n = \mathbf{x}_0 - \sum_{j=1}^{n} \alpha_j \mathbf{s}_j = \mathbf{x}_{n-1} - \alpha_n \mathbf{s}_n, \quad \alpha_n = \frac{(\mathbf{r}_0, \mathbf{s}_n)}{(\mathbf{A}\mathbf{s}_n, \mathbf{s}_n)}. \tag{2.4.3}$$

Формулы (2.4.1)—(2.4.3) лежат в основе метода, а весь последующий материал посвящен его свойствам и улучшению рабочих формул. Умножая (2.4.3) на $\mathbf{A}$ и вычитая $\mathbf{b}$ из обеих частей результата, получаем:

$$\mathbf{r}_n = \mathbf{r}_{n-1} - \alpha_n \mathbf{A}\mathbf{s}_n = \mathbf{r}_0 - \sum_{j=1}^{n} \alpha_j \mathbf{A}\mathbf{s}_j. \tag{2.4.4}$$

Обратимся к скалярному произведению $(\mathbf{r}_n, \mathbf{s}_k)$. Если $k > n$, то слагаемые от суммы в правой части (2.4.4) равны нулю и, таким образом,

$$(\mathbf{r}_n, \mathbf{s}_k) = (\mathbf{r}_0, \mathbf{s}_k), \quad k > n. \tag{2.4.5}$$

А для $n \geq k$ имеем

$$(\mathbf{r}_n, \mathbf{s}_k) = (\mathbf{r}_0, \mathbf{s}_k) - \left(\sum_{j=1}^{n} \alpha_j \mathbf{A}s_j, \mathbf{s}_k \right) = (\mathbf{r}_0, \mathbf{s}_k) - \alpha_k (\mathbf{A}\mathbf{s}_k, \mathbf{s}_k) = (\mathbf{r}_0, \mathbf{s}_k) - (\mathbf{r}_0, \mathbf{s}_k) = 0,$$

т. е.

$$(\mathbf{r}_n, \mathbf{s}_k) = 0, \quad n \geq k. \tag{2.4.6}$$

Наконец, рассмотрим скалярное произведение $(\mathbf{r}_n, \mathbf{r}_k)$. В соответствии с формулой (2.4.2) $\mathbf{r}_k$ является линейной комбинацией векторов $\mathbf{s}_1$, $\mathbf{s}_2$, ..., $\mathbf{s}_{k+1}$. Тогда на основе (2.4.6) легко получаем:

$$(\mathbf{r}_n, \mathbf{r}_k) = 0, \quad n \geq k+1. \tag{2.4.7}$$

Иными словами, векторы невязки, вновь получаемые на каждом шаге, ортогональны всем предыдущим векторам невязки! Процесс заканчивается за m шагов, где m — размер всех векторов, и метод сопряженных градиентов оказывается *точным*, а не итерационным! Осталось несколько улучшить рабочие формулы. Последовательно учитывая формулы (2.4.5), (2.4.2) и (2.4.6), получаем

$$(\mathbf{r}_0, \mathbf{s}_n) = (\mathbf{r}_{n-1}, \mathbf{s}_n) = (\mathbf{r}_{n-1}, \mathbf{r}_{n-1})$$

и для α_n в формуле (2.4.3) получаем новое выражение:

$$\alpha_n = \frac{(\mathbf{r}_{n-1}, \mathbf{r}_{n-1})}{(\mathbf{A}\mathbf{s}_n, \mathbf{s}_n)}. \tag{2.4.8}$$

Хотя аналитически они одинаковы, векторы в числителе (2.4.8) имеют значительно меньшие элементы, что позволяет уменьшить вычислительную погрешность. Теперь уточним выражение для b_{nj} в (2.4.2). Непосредственно с учетом (2.4.4) следует:

$$\mathbf{A}\mathbf{s}_j = \frac{\mathbf{r}_{j-1} - \mathbf{r}_j}{\alpha_j}; \quad b_{nj} = \frac{(\mathbf{A}\mathbf{s}_j, \mathbf{r}_n)}{(\mathbf{A}\mathbf{s}_j, \mathbf{s}_j)} = \frac{(\mathbf{r}_{j-1} - \mathbf{r}_j, \mathbf{r}_n)}{\alpha_j (\mathbf{A}\mathbf{s}_j, \mathbf{s}_j)}.$$

С учетом (2.4.7) для $j < n$ все коэффициенты b_{nj} обращаются в нуль ($b_{nj} = 0,\ j < n$), а для единственного в (2.4.2) ненулевого коэффициента b_{nn} имеем:

$$b_{nn} = \frac{(\mathbf{A}\mathbf{s}_n, \mathbf{r}_n)}{(\mathbf{A}\mathbf{s}_n, \mathbf{s}_n)} = -\frac{(\mathbf{r}_n, \mathbf{r}_n)}{\alpha_n (\mathbf{A}\mathbf{s}_n, \mathbf{s}_n)} = -\frac{(\mathbf{r}_n, \mathbf{r}_n)}{(\mathbf{r}_{n-1}, \mathbf{r}_{n-1})}. \tag{2.4.9}$$

Подведем итоги. Предварительные вычисления состоят в нахождении вектора невязки $\mathbf{r}_0 = \mathbf{A}\mathbf{x}_0 - \mathbf{b}$ по выбранному вектору $\mathbf{x}_0$ и принятии $\mathbf{s}_1 = \mathbf{r}_0$.

Далее по рекуррентным формулам на каждом шаге последовательно вычисляются

$$\alpha_n = \frac{(\mathbf{r}_{n-1}, \mathbf{r}_{n-1})}{(\mathbf{A}\mathbf{s}_n, \mathbf{s}_n)};$$

$$\mathbf{x}_n = \mathbf{x}_{n-1} - \alpha_n \mathbf{s}_n;$$

$$\mathbf{r}_n = \mathbf{r}_{n-1} - \alpha_n \mathbf{A}\mathbf{s}_n;$$

$$b_{nn} = -\frac{(\mathbf{r}_n, \mathbf{r}_n)}{(\mathbf{r}_{n-1}, \mathbf{r}_{n-1})};$$

$$\mathbf{s}_{n+1} = \mathbf{r}_n - b_{nn}\mathbf{s}_n.$$

Для выполнения одного шага нужно одно умножение матрицы на вектор, вычисление двух скалярных произведений и вычисление трех векторов.

Как уже отмечалось, метод сопряженных градиентов является точным. Какова причина такого его превосходства над методами предыдущего раздела? Для ответа на этот вопрос последовательно выполним преобразования:

$$\mathbf{x}_{n+1} = \mathbf{x}_n - \alpha_{n+1}\mathbf{s}_{n+1} = \mathbf{x}_n - \alpha_{n+1}(\mathbf{r}_n - b_{nn}\mathbf{s}_n);$$

$$\frac{\mathbf{x}_{n+1} - \mathbf{x}_n}{\alpha_{n+1}} + b_{nn}\mathbf{s}_n + \mathbf{A}\mathbf{x}_n - \mathbf{b} = 0;$$

$$\frac{\mathbf{x}_{n+1} - \mathbf{x}_n}{\alpha_{n+1}} - \frac{b_{nn}}{\alpha_n}(\mathbf{x}_n - \mathbf{x}_{n-1}) + \mathbf{A}\mathbf{x}_n - \mathbf{b} = 0.$$

Исключив все промежуточные переменные, легко заметить, что разностное уравнение имеет второй порядок, а не первый, как ранее, что и является залогом успеха.

На практике ошибки округления не позволят за m шагов получить точное решение, что будет означать нарушение ортогональности $\mathbf{r}_k$. Однако итерации можно продолжать, компенсируя ошибки округления. Для хорошо обусловленных систем удачный выбор $\mathbf{x}_0$ позволяет значительно уменьшить число итераций, а для плохо обусловленных систем удовлетворительные результаты можно получить увеличением числа итераций. Метод может использоваться для уточнения решения после применения метода Гаусса.

К сожалению, метод сопряженных градиентов предназначен лишь для симметрических положительно определенных матриц. Эту проблему решает трансформация Гаусса, состоящая в умножении системы $\mathbf{A}\mathbf{x} = \mathbf{b}$ слева на

$\mathbf{A}^T$: $\mathbf{A}^T\mathbf{A}\mathbf{x} = \mathbf{A}^T\mathbf{b}$. Матрица $\mathbf{A}^T\mathbf{A}$ положительно определена и симметрична, а система $\mathbf{A}^T\mathbf{A}\mathbf{x} = \mathbf{A}^T\mathbf{b}$ имеет то же решение, что и система $\mathbf{A}\mathbf{x} = \mathbf{b}$. Следует, однако, заметить, что трансформация Гаусса очень сильно ухудшает обусловленность системы, и этот прием не всегда приносит успех.

2.5. Решение проблемы собственных значений

Проблема собственных значений подразумевает отыскание чисел λ_1, λ_2, ..., λ_m (в общем случае комплексных) и соответствующих ненулевых векторов $\mathbf{u}_1$, $\mathbf{u}_2$, ..., $\mathbf{u}_m$, удовлетворяющих уравнению

$$\mathbf{A}\mathbf{u} = \lambda\mathbf{u},$$

где $\mathbf{A}$ — заданная матрица размера $m \times m$. Все методы делятся на две группы. С одной стороны, это методы, предназначенные для решения *полной проблемы собственных значений*, когда определяются все собственные значения матрицы. С другой стороны, это методы, решающие *частичную проблему собственных значений*, когда определяются только некоторые λ_k. Так же, как и при решении систем линейных алгебраических уравнений, первоначально обратимся к вопросу о влиянии на точность результата погрешности исходных данных. В данном случае необходимо ответить на вопрос: "Как погрешность в задании элементов матрицы $\mathbf{A}$ влияет на точность определения собственных значений и собственных векторов?"

2.5.1. Устойчивость проблемы собственных значений

Ограничимся случаем, когда рассматриваемое собственное значение простое. Сохраняя обозначения из *разд. П3.3*, определим λ_k и $\mathbf{u}_k$ как собственные значения и собственные векторы матрицы $\mathbf{A}$:

$$\mathbf{A}\mathbf{u}_k = \lambda\mathbf{u}_k, \tag{2.5.1}$$

а $\mathbf{v}_k$ как собственные векторы транспонированной матрицы $\mathbf{A}^T$:

$$\mathbf{A}^T\mathbf{v}_k = \lambda_k\mathbf{v}_k \quad \text{или} \quad \mathbf{v}_k^T\mathbf{A} = \lambda_k\mathbf{v}_k^T. \tag{2.5.2}$$

Напомним, что в соответствии с внешним видом (2.5.1) и (2.5.2) $\mathbf{u}_k$ часто называют правыми собственными векторами, а $\mathbf{v}_k$ — левыми. Для этих векторов, относящихся к различным собственным значениям, в *разд. П3.3* установлено, что

$$\mathbf{v}_k^T \mathbf{u}_i = 0, \quad i \neq k. \tag{2.5.3}$$

Если допустить небольшие изменения $\Delta\mathbf{A}$ элементов матрицы, то изменения собственного значения $\Delta\lambda_k$ и собственного вектора $\Delta\mathbf{u}_k$ с точностью до элементов второго порядка малости будут удовлетворять линеаризованному уравнению:

$$\Delta\mathbf{A}\mathbf{u}_k + \mathbf{A}\Delta\mathbf{u}_k = \Delta\lambda_k \mathbf{u}_k + \lambda_k \Delta\mathbf{u}_k. \tag{2.5.4}$$

Умножим уравнение (2.5.4) слева на $\mathbf{v}_j^T$

$$v_j^T \Delta\mathbf{A}\mathbf{u}_k + v_j^T \mathbf{A}\Delta\mathbf{u}_k = \Delta\lambda_k v_j^T \mathbf{u}_k + \lambda_k v_j^T \Delta\mathbf{u}_k. \tag{2.5.5}$$

Первоначально полагая в (2.5.5) значение $j = k$, получим

$$\Delta\lambda_k v_k^T \mathbf{u}_k = v_k^T \Delta\mathbf{A}\mathbf{u}_k + \left(v_k^T \mathbf{A} - \lambda_k v_k^T\right)\Delta\mathbf{u}_k = v_k^T \Delta\mathbf{A}\mathbf{u}_k,$$

$$\Delta\lambda_k = \frac{v_k^T \Delta\mathbf{A}\mathbf{u}_k}{v_k^T \mathbf{u}_k}. \tag{2.5.6}$$

Оценивая изменение собственного значения сверху, получаем

$$|\Delta\lambda_k| \le \frac{\|\Delta\mathbf{A}\| \cdot \|v_k\| \cdot \|\mathbf{u}_k\|}{\left|v_k^T \mathbf{u}_k\right|} = C_k \|\Delta\mathbf{A}\|, \quad C_k = \frac{\|v_k\| \cdot \|\mathbf{u}_k\|}{\left|v_k^T \mathbf{u}_k\right|} = \frac{1}{\cos(\varphi_k)}, \tag{2.5.7}$$

где C_k — так называемый *коэффициент перекоса*, а φ_i — угол между собственными векторами $\mathbf{v}_k$ и $\mathbf{u}_k$ матрицы $\mathbf{A}$. (Весьма полезно соотнести это выражение с содержанием положения 2 в *разд. 2.1.*)

Пусть теперь в (2.5.5) значение $j \neq k$. С учетом (2.5.3) имеем

$$v_j^T \Delta\mathbf{A}\mathbf{u}_k = \left(\lambda_k - \lambda_j\right) v_j^T \Delta\mathbf{u}_k, \quad v_j^T \Delta\mathbf{u}_k = \frac{v_j^T \Delta\mathbf{A}\mathbf{u}_k}{\left(\lambda_k - \lambda_j\right)}. \tag{2.5.8}$$

Разложим погрешность собственного вектора $\Delta\mathbf{u}_k$ по системе векторов $\mathbf{u}_j$:

$$\Delta\mathbf{u}_k = \sum_{j=1}^{m} \alpha_{kj} \mathbf{u}_j. \tag{2.5.9}$$

4 Зак. 3418

Так как вектор $\mathbf{u}_k + \Delta\mathbf{u}_k$ определяется с точностью до постоянного множителя, выберем этот множитель так, чтобы диагональный элемент разложения α_{kk} обратился в нуль $(\alpha_{kk} = 0)$. Тогда, подставляя (2.5.9) в (2.5.8), с учетом (2.5.3) для α_{kj} получаем:

$$\alpha_{kj} = \frac{v_j^T \Delta\mathbf{A}\mathbf{u}_k}{\left((\lambda_k - \lambda_j) v_j^T \mathbf{u}_j\right)}. \qquad (2.5.10)$$

После подстановки (2.5.10) в (2.5.9) вектор погрешности собственного вектора может быть оценен сверху

$$\|\Delta\mathbf{u}_k\| \le \|\Delta\mathbf{A}\| \cdot \|\mathbf{u}_k\| \cdot \sum_{j \ne k} \frac{\|\mathbf{v}_j\| \cdot \|\mathbf{u}_j\|}{|\lambda_k - \lambda_j| \cdot |v_j^T \mathbf{u}_j|}$$

или

$$\frac{\|\Delta\mathbf{u}_k\|}{\|\mathbf{u}_k\|} \le \|\Delta\mathbf{A}\| \cdot \sum_{j \ne k} \frac{C_j}{|\lambda_k - \lambda_j|}. \qquad (2.5.11)$$

Неравенства (2.5.7) и (2.5.11) позволяют сделать следующие выводы. Чтобы изменение собственного значения было соизмеримо с величиной $\|\Delta\mathbf{A}\|$, отвечающий ему коэффициент перекоса должен быть не слишком велик. Так для симметрической матрицы, когда векторы $\mathbf{u}_k$ и $\mathbf{v}_k$ совпадают и $C_k = 1$, неравенство (2.5.7) приобретает весьма простой вид:

$$|\Delta\lambda_k| \le \|\Delta\mathbf{A}\|.$$

Изменение собственного вектора аналогичным образом зависит уже от *всех* коэффициентов перекоса. При этом легко заметить, что, даже если λ_k — простое собственное значение, наличие близких к нему собственных значений из-за малой величины $|\lambda_k - \lambda_j|$ может существенно повлиять на изменение $\|\Delta\mathbf{u}_k\|$.

2.5.2. Частичная проблема собственных значений. Степенной метод

Степенной метод (или прямые итерации) предназначен для нахождения максимального по модулю собственного значения и соответствующего собственного вектора. Идея, лежащая в основе этого метода, широко использу-

ется и в других более эффективных алгоритмах. Пусть все собственные значения λ_k матрицы $\mathbf{A}$ различны и упорядочены следующим образом:

$$|\lambda_1| > |\lambda_2| \geq |\lambda_3| \geq |\lambda_4| \geq \ldots \geq |\lambda_m|.$$

Зададимся некоторым начальным вектором $\mathbf{x}_0$, который разложим по собственным векторам $\mathbf{u}_k$ матрицы $\mathbf{A}$:

$$\mathbf{x}_0 = \alpha_1 \mathbf{u}_1 + \alpha_2 \mathbf{u}_2 + \ldots + \alpha_m \mathbf{u}_m, \tag{2.5.12}$$

предполагая, что $\alpha_1 \neq 0$. Так как собственные векторы определяются с точностью до постоянного множителя, все коэффициенты α_k в (2.5.12) можно принять единичными. Однако для удобства мы сохраним приведенный вид этой формулы.

В ходе применения степенного метода строится последовательность векторов $\mathbf{x}_k$:

$$\mathbf{x}_{k+1} = \mathbf{A}\mathbf{x}_k, \tag{2.5.13}$$

$$\mathbf{x}_{k+1} = \mathbf{A}^k \mathbf{x}_0 = \alpha_1 \lambda_1^k \mathbf{u}_1 + \alpha_2 \lambda_2^k \mathbf{u}_2 + \ldots + \alpha_m \lambda_m^k \mathbf{u}_m =$$
$$= \lambda_1^k \left(\alpha_1 \mathbf{u}_1 + \alpha_2 \left(\frac{\lambda_2}{\lambda_1} \right)^k \mathbf{u}_2 + \ldots + \alpha_m \left(\frac{\lambda_m}{\lambda_1} \right)^k \mathbf{u}_m \right).$$

Для наименьшего угла φ между векторами $\mathbf{x}_k$ и $\mathbf{u}_1$ получаем:

$$\cos\varphi = \frac{|(\mathbf{x}_k, \mathbf{u}_1)|}{\|\mathbf{x}_k\| \cdot \|\mathbf{u}_1\|} = \frac{\left| \|\mathbf{u}_1\|^2 + (\lambda_2/\lambda_1)^k (\mathbf{u}_2, \mathbf{u}_1) + \ldots + (\lambda_m/\lambda_1)^k (\mathbf{u}_m, \mathbf{u}_1) \right|}{\|\mathbf{u}_1\| \cdot \left\| \mathbf{u}_1 + (\lambda_2/\lambda_1)^k \mathbf{u}_2 + \ldots + (\lambda_m/\lambda_1)^k \mathbf{u}_m \right\|}.$$

Так как $|\lambda_2/\lambda_1| < 1$, $\cos\varphi \to 1$ при $k \to \infty$. Иными словами, вектор $\mathbf{x}_k$ сходится к вектору $\mathbf{u}_1$, отвечающему собственному значению λ_1. При больших значениях k имеет место приближенное равенство

$$\mathbf{x}_{k+1} = \mathbf{A}^{k+1} \mathbf{x}_0 \approx \lambda_1^{k+1} \mathbf{u}_1 \approx \lambda_1 \mathbf{x}_k. \tag{2.5.14}$$

Это позволяет вычислять приближенно λ_1 как отношение каких-либо компонентов векторов $\mathbf{x}_{k+1}$ и $\mathbf{x}_k$

$$\lambda_1 \approx \frac{x_{k+1}^{(j)}}{x_k^{(j)}}$$

или как частное

$$\lambda_1 \approx \frac{(\mathbf{x}_{k+1}, \mathbf{b})}{(\mathbf{x}_k, \mathbf{b})}, \tag{2.5.15}$$

где $\mathbf{b}$ — произвольный вектор. В частности, в роли $\mathbf{b}$ может использоваться $\mathbf{x}_k$:

$$\lambda_1 \approx \frac{(\mathbf{x}_{k+1}, \mathbf{x}_k)}{(\mathbf{x}_k, \mathbf{x}_k)}.$$

При $|\lambda_1| < 1$ последовательность $\mathbf{x}_k$ сходится к нулю, а при $|\lambda_1| > 1$ она неограниченно возрастает. Поэтому целесообразно нормировать вектор $\mathbf{x}_k$ на каждой итерации. В итоге рабочие формулы (2.5.13) и (2.5.15) дополняются условием нормировки, и на ($k+1$)-м шаге используется вектор $\mathbf{x}_k / \|\mathbf{x}_k\|$.

Если начальное приближение $\mathbf{x}_0$ выбрано неудачно, так, что $\alpha_1 = 0$, то теоретически итерационный процесс должен сходиться к λ_2. Однако на практике нередко наблюдается следующий эффект. Сначала процесс начинает сходиться к λ_2, а затем все же сходится к λ_1. Связано это с тем, что из-за ошибок округлений в разложении $\mathbf{x}_k$ по собственным векторам возникает ненулевой коэффициент α_1. Слагаемое, относящееся к $\mathbf{u}_1$, сначала очень малое, начинает заметно расти с ростом k.

Следует отметить, что рассмотренная идея легко распространяется на случай, когда максимальное по модулю собственное значение является комплексным или кратным. Имеется также модификация, позволяющая найти одновременно сразу группу собственных значений, превышающих остальные по модулю.

Обратный степенной метод (или обратные итерации) сводится к степенному методу, примененному к обратной матрице $\mathbf{A}^{-1}$. Рабочие формулы метода имеют вид:

$$\mathbf{A}\mathbf{x}_{k+1} = \mathbf{x}_k, \tag{2.5.16}$$

$$\lambda_m \approx \frac{(\mathbf{x}_k, \mathbf{b})}{(\mathbf{x}_{k+1}, \mathbf{b})}. \tag{2.5.17}$$

Максимальное по модулю собственное значение матрицы $\mathbf{A}^{-1}$ равно $(1/\lambda_m)$. Аналогично предыдущему

$$\mathbf{x}_{k+1} = \mathbf{A}^{-k-1}\mathbf{x}_0 \approx \left(\frac{1}{\lambda_m}\right)^{k+1} \cdot \mathbf{u}_1 \approx \frac{1}{\lambda_m}\mathbf{x}_k, \tag{2.5.18}$$

откуда и следует (2.5.17). Если скорость сходимости прямых итераций определяется величиной $|\lambda_2/\lambda_1|$, то в случае обратных итераций эта скорость зависит от $|\lambda_m/\lambda_{m-1}|$. В поиске обратной матрицы нет необходимости, достаточно на каждом шаге решать линейную систему (2.5.16). Трудоемкость расчетов резко снижается, если *однократно* определять LU-разложение матрицы **A** (например, программой DECOMP), а затем на каждой итерации решать две системы с треугольными матрицами (например, программой SOLVE). Как и для прямых итераций, рабочие формулы (2.5.16) и (2.5.17) необходимо дополнить условиями нормировки и использовать вектор $\mathbf{x}_k/\|\mathbf{x}_k\|$.

Метод обратных итераций может быть распространен и на случай поиска собственного значения, ближайшего к некоторой величине μ. В этом случае он называется *обратный степенной метод со сдвигом* (или метод Виландта). Очень часто он используется, когда собственное значение λ_k уже найдено с некоторой погрешностью, и требуется лишь уточнить λ_k и найти соответствующий собственный вектор. В этом случае вместо (2.5.16) предлагается использовать следующую формулу:

$$(\mathbf{A}-\mu\mathbf{E})\mathbf{x}_{k+1}=\mathbf{x}_k. \qquad (2.5.19)$$

Легко заметить, что собственное значение, ближайшее к μ, является минимальным по модулю собственным значением матрицы $\mathbf{A}-\mu\mathbf{E}$. Здесь также принято сокращать объем вычислений однократным предварительным приведением исходной матрицы **A** к одной из специальных форм (трехдиагональной, хессенберговой и т. п.), сокращающей решение (2.5.19) на каждой итерации. Сдвиг иногда изменяется на каждом шаге:

$$(\mathbf{A}-\mu_k\mathbf{E})\mathbf{x}_{k+1}=\mathbf{x}_k. \qquad (2.5.20)$$

В роли μ_k для ускорения сходимости может быть использовано очередное приближение к искомому собственному значению. В такой ситуации необходимо лишь учитывать, что матрица системы становится очень плохо обусловленной.

2.5.3. Полная проблема собственных значений. QR-алгоритм

QR-алгоритм является самым популярным современным алгоритмом для решения полной проблемы собственных значений (нахождения всех собствен-

ных значений матрицы). Его авторами являются В. Н. Кублановская и Дж. Френсис (Francis J. G. F). До изложения основных формул выполним некоторую подготовительную работу.

Определение. Квадратная матрица называется *матрицей Хессенберга*, если ее элементы a_{ik} равны нулю для $i > k + 1$.

Матрица Хессенберга имеет "почти треугольный" вид:

$$\begin{pmatrix} * & * & * & * & * & * \\ * & * & * & * & * & * \\ 0 & * & * & * & * & * \\ 0 & 0 & * & * & * & * \\ 0 & 0 & 0 & * & * & * \\ 0 & 0 & 0 & 0 & * & * \end{pmatrix}.$$

Иными словами, это верхняя треугольная матрица и еще одна поддиагональ. Остальные элементы равны нулю. Здесь и в дальнейшем символом "*" обозначаются элементы матрицы, которые в общем случае отличны от нуля.

Определение. *Преобразованием Хаусхолдера* называется преобразование со следующей матрицей

$$\mathbf{H} = \mathbf{E} - 2\frac{\mathbf{v}\mathbf{v}^T}{\|\mathbf{v}\|^2}, \qquad (2.5.21)$$

где $\mathbf{v}$ — вектор, порождающий преобразование. Его часто нормируют и используют вектор единичной длины: $\mathbf{u} = \mathbf{v}/\|\mathbf{v}\|$. Тогда матрица $\mathbf{H}$ приобретает вид:

$$\mathbf{H} = \mathbf{E} - 2\mathbf{u}\mathbf{u}^T, \quad \|\mathbf{u}\| = 1. \qquad (2.5.22)$$

Здесь везде и в дальнейшем используется сферическая норма вектора $\|\mathbf{u}\|^2 = \mathbf{u}^T\mathbf{u} = (\mathbf{u}, \mathbf{u})$.

Свойства преобразования Хаусхолдера

1. Матрица преобразования Хаусхолдера является симметрической.

$$\mathbf{H}^T = \mathbf{E} - 2\mathbf{u}\mathbf{u}^T = \mathbf{H}.$$

2. Матрица преобразования Хаусхолдера является ортогональной, т. е. $\mathbf{H}^T\mathbf{H} = \mathbf{E}$.

$$\mathbf{H}^T\mathbf{H} = \mathbf{H}^2 = \left(\mathbf{E} - 2\mathbf{u}\mathbf{u}^T\right)\left(\mathbf{E} - 2\mathbf{u}\mathbf{u}^T\right) = \mathbf{E} - 4\mathbf{u}\mathbf{u}^T + 4\mathbf{u}\mathbf{u}^T\mathbf{u}\mathbf{u}^T = \mathbf{E}.$$

3. Пусть $\mathbf{e}_k$ — k-й столбец матрицы $\mathbf{E}$, $\mathbf{v} = \mathbf{x} + \|\mathbf{x}\| \cdot \mathbf{e}_k$. Тогда $\mathbf{H}\mathbf{x} = -\|\mathbf{x}\| \cdot \mathbf{e}_k$.

Доказательство.

$$\mathbf{H}\mathbf{x} = \mathbf{E}\mathbf{x} - 2\frac{\mathbf{v}\mathbf{v}^T\mathbf{x}}{\|\mathbf{v}\|^2} = \mathbf{x} - \left(\mathbf{x} + \|\mathbf{x}\| \cdot \mathbf{e}_k\right) \cdot \frac{2 \cdot \left(\mathbf{x} + \|\mathbf{x}\| \cdot \mathbf{e}_k\right)^T \cdot \mathbf{x}}{\left(\mathbf{x} + \|\mathbf{x}\| \cdot \mathbf{e}_k\right)^T \cdot \left(\mathbf{x} + \|\mathbf{x}\| \cdot \mathbf{e}_k\right)} =$$

$$= \mathbf{x} - \left(\mathbf{x} + \|\mathbf{x}\| \cdot \mathbf{e}_k\right) \cdot \frac{2\left(\mathbf{x} + \|\mathbf{x}\| \cdot \mathbf{e}_k\right)^T \mathbf{x}}{\mathbf{x}^T\mathbf{x} + \|\mathbf{x}\| \cdot \mathbf{x}^T\mathbf{e}_k + \|\mathbf{x}\| \cdot \mathbf{e}_k^T\mathbf{x} + \|\mathbf{x}\|^2 \cdot \mathbf{e}_k^T\mathbf{e}_k} =$$

$$= \mathbf{x} - \left(\mathbf{x} + \|\mathbf{x}\| \cdot \mathbf{e}_k\right) \cdot \frac{2\|\mathbf{x}\|^2 + 2\|\mathbf{x}\| \cdot \mathbf{e}_k^T\mathbf{x}}{2\|\mathbf{x}\|^2 + 2\|\mathbf{x}\| \cdot \mathbf{e}_k^T\mathbf{x}} = -\|\mathbf{x}\| \cdot \mathbf{e}_k.$$

Произвольная квадратная матрица может быть приведена к форме Хессенберга с помощью преобразования подобия Хаусхолдера, сохраняющего собственные значения исходной матрицы.

Пусть исходная матрица $\mathbf{A}$ имеет размер $m \times m$. Выберем в качестве вектора $\mathbf{x}_1$ первый столбец матрицы $\mathbf{A}$ без первого элемента: $\mathbf{x}_1 = (a_{21}, a_{31}, ..., a_{m1})^T$.

Сформируем вектор $\mathbf{v}_1 = \mathbf{x}_1 + \|\mathbf{x}_1\| \cdot \mathbf{e}_1$ и на его основе построим матрицу Хаусхолдера $\mathbf{H}_1$. Векторы $\mathbf{x}_1$, $\mathbf{v}_1$, $\mathbf{e}_1$ имеют размер $m-1$, а матрица $\mathbf{H}_1$ — размер $(m-1)\times(m-1)$ соответственно. Уже на основе $\mathbf{H}_1$ определим ортогональную матрицу $\mathbf{U}_1$ размера $m \times m$, первая строка и первый столбец которой совпадают с единичной матрицей $\mathbf{E}$, а остальные элементы равны элементам $\mathbf{H}_1$:

$$\mathbf{U}_1 = \begin{pmatrix} 1 & 0 & 0 & \ldots & 0 \\ 0 & \mathbf{H}_1 & & & \\ 0 & & & & \\ \ldots & & & & \\ 0 & & & & \end{pmatrix} = \mathbf{U}_1^{-1} = \mathbf{U}_1^T.$$

После умножения матрицы $\mathbf{A}$ слева на $\mathbf{U}_1$ в силу свойства 3 первый столбец приобретает вид столбца матрицы Хессенберга:

$$\mathbf{U}_1\mathbf{A} = \begin{pmatrix} a_{11} & * & * & \dots & * \\ -\|\mathbf{x}_1\| & * & * & \dots & * \\ 0 & * & * & \dots & * \\ \dots & \dots & \dots & \dots & \dots \\ 0 & * & * & \dots & * \end{pmatrix},$$

а умножение результата на $\mathbf{U}_1$ справа не изменяет структуры первого столбца. Таким образом, после первого шага от матрицы $\mathbf{A}$ переходим к подобной матрице $\mathbf{A}_1 = \mathbf{U}_1\mathbf{A}\mathbf{U}_1$ с нужным видом первого столбца и теми же собственными значениями.

Второй шаг выглядит аналогично. Сначала строим вектор $\mathbf{x}_2$ размера $m-2$, как второй столбец матрицы $\mathbf{A}_1$ без первых двух элементов, и вектор $\mathbf{v}_2$:

$$\mathbf{x}_2 = \left(\overset{*}{a}_{32}, \overset{*}{a}_{42}, \dots, \overset{*}{a}_{m2}\right)^T, \quad \mathbf{v}_2 = \mathbf{x}_2 + \|\mathbf{x}_2\| \cdot \mathbf{e}_1.$$

Здесь вектор $\mathbf{e}_1$ — по-прежнему первый столбец единичной матрицы $\mathbf{E}$, но уже размера $m-2$. На основе $\mathbf{v}_2$ получаем матрицу Хаусхолдера $\mathbf{H}_2$ размера $(m-2)\times(m-2)$. Аналогично предыдущему шагу определим ортогональную матрицу $\mathbf{U}_2$ размера $m \times m$, первые две строки и два столбца которой совпадают с единичной матрицей $\mathbf{E}$, а остальные элементы равны элементам $\mathbf{H}_2$:

$$\mathbf{U}_2 = \begin{pmatrix} 1 & 0 & 0 & \dots & 0 \\ 0 & 1 & 0 & \dots & 0 \\ \dots & \dots & \mathbf{H}_2 & & \\ \dots & \dots & & & \\ 0 & 0 & & & \end{pmatrix} = \mathbf{U}_2^{-1} = \mathbf{U}_2^T.$$

Подобное преобразование с этой матрицей позволяет получить матрицу $\mathbf{A}_2 = \mathbf{U}_2\mathbf{A}_1\mathbf{U}_2$, первые два столбца которой имеют вид столбцов матрицы Хессенберга:

$$\mathbf{A}_2 = \mathbf{U}_2\mathbf{A}_1\mathbf{U}_2 = \mathbf{U}_2\left(\mathbf{U}_1\mathbf{A}\mathbf{U}_1\right) = \begin{pmatrix} * & * & * & * & * & * \\ * & * & * & * & * & * \\ 0 & * & * & * & * & * \\ 0 & 0 & * & * & * & * \\ . & . & . & . & . & . \\ 0 & 0 & * & * & * & * \end{pmatrix}.$$

Продолжение этого процесса позволяет за $m-2$ шага получить матрицу $\tilde{\mathbf{A}}$, имеющую форму Хессенберга и сохраняющую те же собственные значения, что и исходная матрица $\mathbf{A}$:

$$\tilde{\mathbf{A}} = \left(\mathbf{U}_{m-2}\mathbf{U}_{m-3}...\mathbf{U}_2\mathbf{U}_1\right)\mathbf{A}\left(\mathbf{U}_1\mathbf{U}_2...\mathbf{U}_{m-3}\mathbf{U}_{m-2}\right) = \mathbf{S}^{-1}\mathbf{A}\mathbf{S} = \mathbf{S}^{\mathrm{T}}\mathbf{A}\mathbf{S},$$

$$\mathbf{S} = \mathbf{U}_1\mathbf{U}_2...\mathbf{U}_{m-3}\mathbf{U}_{m-2}.$$

Легко заметить, что если матрица $\mathbf{A}$ была симметрической, то это свойство сохраняется и у $\tilde{\mathbf{A}}$, т. е. она оказывается трехдиагональной.

Преобразование Хаусхолдера не является единственным способом, приводящим матрицу к форме Хессенберга. Этой же цели можно достигнуть, например, с помощью преобразования Гивенса.

Определение. *Преобразованием Гивенса* называется преобразование с матрицей (2.5.23), называемой также *матрицей вращения*. Эта матрица отличается от единичной $\mathbf{E}$ только четырьмя элементами, стоящими на пересечении двух строк и двух столбцов с номерами k и j:

$$t_{kk} = t_{jj} = c; \quad t_{jk} = -t_{kj} = s; \quad c^2 + s^2 = 1.$$

Непосредственным перемножением убеждаемся, что матрица вращения является ортогональной:

$$\mathbf{T}_{kj}^{-1} = \mathbf{T}_{kj}^{T}; \quad \mathbf{T}_{kj}^{T} \cdot \mathbf{T}_{kj} = \mathbf{E}.$$

$$\mathbf{T}_{kj} = \begin{pmatrix} 1 & & & & & & & \\ & \cdots & & & & & & \\ & & c & \cdots & \cdots & \cdots & -s & \\ & & \cdots & \cdots & & & & \\ & & \cdots & & 1 & & & \\ & & \cdots & & & \cdots & & \\ & & s & \cdots & \cdots & \cdots & c & \\ & & & & & & & \cdots \\ & & & & & & & & 1 \end{pmatrix}. \quad (2.5.23)$$

Обозначим за $\mathbf{A}_{(k)}$ k-й столбец матрицы $\mathbf{A}$, а за $\mathbf{A}^{(k)}$ k-ю строку этой матрицы и умножим $\mathbf{A}$ на $\mathbf{T}_{kj}$ ($\mathbf{B} = \mathbf{A}\mathbf{T}_{kj}$). Матрицы $\mathbf{A}$ и $\mathbf{B}$ отличаются лишь двумя столбцами:

$$\mathbf{B}_{(k)} = c\mathbf{A}_{(k)} + s\mathbf{A}_{(j)}; \quad \mathbf{B}_{(j)} = -s\mathbf{A}_{(k)} + c\mathbf{A}_{(j)}. \quad (2.5.24)$$

Теперь матрицу $\mathbf{T}_{kj}^T$ умножим на $\mathbf{B}$ $\left(\mathbf{G} = \mathbf{T}_{kj}^T\mathbf{B}\right)$. Матрицы $\mathbf{G}$ и $\mathbf{B}$, в свою очередь, отличаются лишь двумя строками:

$$\mathbf{G}^{(k)} = c\mathbf{B}^{(k)} + s\mathbf{B}^{(j)}; \qquad \mathbf{G}^{(j)} = -s\mathbf{B}^{(k)} + c\mathbf{B}^{(j)}. \quad (2.5.25)$$

Таким образом, в матрице $\mathbf{G} = \mathbf{T}_{kj}^T\mathbf{A}\mathbf{T}_{kj}$ по сравнению с $\mathbf{A}$ изменяются лишь две строки и два столбца с номерами k и j. Обозначим элементы матрицы $\mathbf{G}$ как g_{ps}. Принимая во всех дальнейших рассуждениях $1 < j < k$, выберем свободные параметры c и s так, чтобы $g_{k,\,j-1} = 0$.

$$g_{k,\,j-1} = cb_{k,\,j-1} + sb_{j,\,j-1} = ca_{k,\,j-1} + sa_{j,\,j-1}.$$

Последнее равенство очевидно, т. к. столбец с номером $j-1$ в матрицах $\mathbf{B}$ и $\mathbf{A}$ один и тот же. Выполняя требование $g_{k,\,j-1} = 0$, получаем следующие выражения для c и s:

$$\frac{s}{c} = -\frac{a_{k,\,j-1}}{a_{j,\,j-1}}; \; s = -\frac{a_{k,j-1}}{\sqrt{a_{k,j-1}^2 + a_{j,j-1}^2}}; \; c = \frac{a_{j,j-1}}{\sqrt{a_{k,j-1}^2 + a_{j,j-1}^2}}. \quad (2.5.26)$$

Проводя последовательно подобные преобразования с матрицами $\mathbf{T}_{32}$, $\mathbf{T}_{42}$, ..., $\mathbf{T}_{m2}$, обеспечиваем вид первого столбца, как у матрицы Хессенберга. Желаемая форма второго столбца достигается использованием матриц $\mathbf{T}_{43}$, $\mathbf{T}_{53}$, ..., $\mathbf{T}_{m3}$ и т. д. Выполнив суммарно $\frac{(m-2)(m-1)}{2}$ аналогичных преобразований, получаем матрицу в форме Хессенберга. Легко убедиться в том, что каждое последующее преобразование сохраняет ранее полученные нулевые элементы. Хотя для достижения результата количество требуемых преобразований Гивенса заметно больше числа преобразований Хаусхолдера, вычисления по формулам (2.5.24)—(2.5.26) сводятся всего лишь к изменению двух строк и столбцов.

Теперь обратимся к построению QR-разложения матрицы.

Определение. QR-разложением матрицы $\mathbf{A}$ называется ее представление в виде $\mathbf{A}=\mathbf{QR}$, где $\mathbf{Q}$ — ортогональная, а $\mathbf{R}$ — верхняя треугольная матрицы.

QR-разложение матрицы может быть построено различными способами, например, на основе преобразований Хаусхолдера. При этом план построения весьма похож на процедуру приведения матрицы к форме Хессенберга.

Пусть исходная матрица $\mathbf{A}$ имеет размер $m \times m$. Выберем в качестве вектора $\mathbf{x}_1$ первый столбец матрицы $\mathbf{A}$: $\mathbf{x}_1 = (a_{11}, a_{21}, \ldots, a_{m1})^T$.

Сформируем вектор $\mathbf{v}_1 = \mathbf{x}_1 + \|\mathbf{x}_1\| \cdot \mathbf{e}_1$ и на его основе построим матрицу Хаусхолдера $\mathbf{H}_1$. Векторы $\mathbf{x}_1$, $\mathbf{v}_1$, $\mathbf{e}_1$ имеют размер m, а матрица $\mathbf{H}_1$ — размер $m \times m$ соответственно. На основе $\mathbf{H}_1$ определим ортогональную матрицу $\mathbf{U}_1 = \mathbf{H}_1$ и умножим матрицу $\mathbf{A}$ слева на $\mathbf{U}_1$. Первый столбец получившейся матрицы в соответствии со свойством 3 преобразования Хаусхолдера приобретает вид первого столбца верхней треугольной матрицы:

$$\mathbf{U}_1\mathbf{A} = \begin{pmatrix} -\|\mathbf{x}_1\| & * & * & & * \\ 0 & * & * & & * \\ 0 & * & * & & * \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ 0 & * & * & & * \end{pmatrix}.$$

На втором шаге сначала строим вектор $\mathbf{x}_2$ размера $m-1$, как второй столбец матрицы $\mathbf{U}_1\mathbf{A}$ без первого элемента, и вектор $\mathbf{v}_2$:

$$\mathbf{x}_2 = (\overset{*}{a}_{22}, \overset{*}{a}_{32}, ..., \overset{*}{a}_{m2})^T, \quad \mathbf{v}_2 = \mathbf{x}_2 + \|\mathbf{x}_2\| \cdot \mathbf{e}_1.$$

Здесь вектор $\mathbf{e}_1$ по-прежнему первый столбец единичной матрицы $\mathbf{E}$, но уже размера $m-1$. На основе $\mathbf{v}_2$ получаем матрицу Хаусхолдера $\mathbf{H}_2$ размера $(m-1)\times(m-1)$. Аналогично предыдущему шагу определим ортогональную матрицу $\mathbf{U}_2$ размера $m \times m$, первые строка и столбец которой совпадают с единичной матрицей $\mathbf{E}$, а остальные элементы равны элементам $\mathbf{H}_2$:

$$\mathbf{U}_2 = \begin{pmatrix} 1 & 0 & 0 & ... & 0 \\ 0 & \mathbf{H}_2 & & & \\ 0 & & & & \\ ... & & & & \\ 0 & & & & \end{pmatrix}.$$

Умножая матрицу $\mathbf{U}_1\mathbf{A}$ справа на $\mathbf{U}_2$, обеспечиваем уже два столбца верхней треугольной матрицы:

$$\mathbf{U}_2\mathbf{U}_1\mathbf{A} = \begin{pmatrix} * & * & * & * & * & * \\ 0 & * & * & * & * & * \\ 0 & 0 & * & * & * & * \\ 0 & 0 & * & * & * & * \\ . & . & . & . & . & . \\ 0 & 0 & * & * & * & * \end{pmatrix}.$$

Продолжение этого процесса позволяет за $m-1$ шаг получить верхнюю треугольную матрицу $\mathbf{R}$:

$$(\mathbf{U}_{m-1}\mathbf{U}_{m-2}...\mathbf{U}_2\mathbf{U}_1)\mathbf{A} = \mathbf{R}; \quad \mathbf{A} = \mathbf{QR};$$

$$\mathbf{Q} = (\mathbf{U}_{m-1}\mathbf{U}_{m-2}...\mathbf{U}_2\mathbf{U}_1)^{-1} = (\mathbf{U}_{m-1}\mathbf{U}_{m-2}...\mathbf{U}_2\mathbf{U}_1)^T = \mathbf{U}_1\mathbf{U}_2...\mathbf{U}_{m-2}\mathbf{U}_{m-1}.$$

Ортогональность матрицы $\mathbf{Q}$ непосредственно следует из симметричности и ортогональности матриц $\mathbf{U}_k$.

Приступим к изложению QR-алгоритма решения полной проблемы собственных значений.

QR-алгоритм

Грубая схема алгоритма выглядит следующим образом. Обозначим $\mathbf{A}_0 = \mathbf{A}$. На шаге с номером k выполняется QR-разложение матрицы $\mathbf{A}_k$. Затем матрицы $\mathbf{Q}_k$ и $\mathbf{R}_k$ перемножаются в обратном порядке для получения $\mathbf{A}_{k+1}$. При этом матрицы $\mathbf{A}_k$ и $\mathbf{A}_{k+1}$ оказываются подобными, т. е. собственные значения исходной матрицы в результате такого преобразования сохраняются.

$$\mathbf{A}_k = \mathbf{Q}_k\mathbf{R}_k; \quad \mathbf{A}_{k+1} = \mathbf{R}_k\mathbf{Q}_k; \quad \mathbf{A}_{k+1} = \mathbf{Q}_k^{-1}\mathbf{A}_k\mathbf{Q}_k = \mathbf{Q}_k^T\mathbf{A}_k\mathbf{Q}_k. \qquad (2.5.27)$$

Пусть первоначально собственные значения матрицы $\mathbf{A}$ различны. Тогда, как это было показано авторами метода, с ростом числа шагов сумма модулей элементов $\mathbf{A}_k$, расположенных ниже главной диагонали, убывает и в пределе при $k \to \infty$ стремится к нулю. Матрица $\mathbf{A}_k$ становится в пределе верхней треугольной, а ее диагональные элементы стремятся к собственным значениям исходной матрицы. Если среди собственных значений $\mathbf{A}$ есть комплексно-сопряженные или кратные, то $\mathbf{A}_k$ в пределе становится блочной квазитреугольной:

$$\mathbf{A}_k \to \begin{pmatrix} \mathbf{A}_{11} & \mathbf{A}_{12} & \mathbf{A}_{13} & \dots & \mathbf{A}_{1M} \\ \mathbf{0} & \mathbf{A}_{22} & \mathbf{A}_{23} & \dots & \mathbf{A}_{2M} \\ \mathbf{0} & \mathbf{0} & \mathbf{A}_{33} & \dots & \mathbf{A}_{3M} \\ . & . & . & . & . \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \dots & \mathbf{A}_{MM} \end{pmatrix}.$$

Здесь $\mathbf{A}_{ii}$ — квадратные подматрицы (матричные клетки), а $\mathbf{A}_{ij} = 0$ для всех $i > j$. Собственные значения λ_k исходной матрицы $\mathbf{A}$ определяются собственными значениями всех диагональных клеток. Если вещественные λ_k не являются кратными, то им отвечают диагональные клетки первого порядка 1×1. Комплексно-сопряженной паре $\lambda_{k,\,k+1}$ отвечает диагональная клетка 2×2, а если λ_k имеет кратность s, то соответствующий размер $s\times s$ имеет и диагональная клетка.

Непосредственная реализация QR-алгоритма в форме (2.5.27) сталкивается с целым рядом затруднений. С одной стороны, алгоритм сходится относительно медленно, а с другой стороны, на каждом шаге необходимо осуществлять трудоемкую операцию QR-разложения и последующее умножение матриц. Сокращение объема вычислений при реализации алгоритма на практике реализуется двумя путями: уменьшением количества итераций, т. е. повышени-

ем скорости сходимости, и снижением трудоемкости одного шага. В первом случае используется алгоритм со сдвигами, а во втором случае матрица предварительно приводится к форме Хессенберга.

Алгоритм со сдвигами

Как и в случае обратного степенного метода, использование сдвигов заметно повышает скорость сходимости. Если известно приближение τ_k к некоторому собственному значению матрицы $\mathbf{A}$, то рабочие формулы алгоритма со сдвигами приобретают следующий вид:

$$\mathbf{A}_k - \tau_k \mathbf{E} = \mathbf{Q}_k \mathbf{R}_k; \quad \mathbf{A}_{k+1} = \mathbf{R}_k \mathbf{Q}_k + \tau_k \mathbf{E}. \qquad (2.5.28)$$

Нетрудно заметить, что каждая итерация по-прежнему сохраняет собственные значения матрицы $\left(\mathbf{A}_{k+1} = \mathbf{Q}_k^{-1} \mathbf{A}_k \mathbf{Q}_k = \mathbf{Q}_k^T \mathbf{A}_k \mathbf{Q}_k\right)$. Существует несколько стратегий сдвига, многие из которых практически безупречно работают на практике в течение многих лет, но глобальная сходимость для них не установлена.

Предварительное приведение матрицы к форме Хессенберга

Для снижения объема вычислений исходная матрица $\mathbf{A}_0 = \mathbf{A}$ размера $m \times m$ предварительно однократно приводится к форме Хессенберга, что значительно снижает трудоемкость итерации QR-алгоритма. Если в случае матрицы общего вида количество операций умножения и сложения для реализации одного шага по формулам (2.5.27) пропорционально кубу размера $\left(\sim m^3\right)$, то для матрицы Хессенберга это число пропорционально всего лишь квадрату размера $\left(\sim m^2\right)$. Решающим фактором является то, что последующие шаги алгоритма сохраняют форму Хессенберга для $\mathbf{A}_k$. Покажем справедливость этого утверждения.

Пусть матрица $\mathbf{A}_0$ имеет форму Хессенберга:

$$\mathbf{A}_0 = \begin{pmatrix} * & * & * & * \\ * & * & * & * \\ 0 & * & * & * \\ 0 & 0 & * & * \end{pmatrix}.$$

Тогда $\mathbf{Q}_0 = \mathbf{A}_0\mathbf{R}_0^{-1}$ также имеет такой же вид, в чем легко убедиться непосредственным умножением:

$$\mathbf{Q}_0 = \mathbf{A}_0\mathbf{R}_0^{-1} = \begin{pmatrix} * & * & * & * \\ * & * & * & * \\ 0 & * & * & * \\ 0 & 0 & * & * \end{pmatrix} \cdot \begin{pmatrix} * & * & * & * \\ 0 & * & * & * \\ 0 & 0 & * & * \\ 0 & 0 & 0 & * \end{pmatrix} = \begin{pmatrix} * & * & * & * \\ * & * & * & * \\ 0 & * & * & * \\ 0 & 0 & * & * \end{pmatrix}.$$

Теперь, перемножая $\mathbf{Q}_0$ и $\mathbf{R}_0$ в обратном порядке, убеждаемся в том, что $\mathbf{A}_1$ сохраняет форму $\mathbf{A}_0$:

$$\mathbf{A}_1 = \mathbf{R}_0\mathbf{Q}_0 \begin{pmatrix} * & * & * & * \\ 0 & * & * & * \\ 0 & 0 & * & * \\ 0 & 0 & 0 & * \end{pmatrix} \cdot \begin{pmatrix} * & * & * & * \\ * & * & * & * \\ 0 & * & * & * \\ 0 & 0 & * & * \end{pmatrix} = \begin{pmatrix} * & * & * & * \\ * & * & * & * \\ 0 & * & * & * \\ 0 & 0 & * & * \end{pmatrix}.$$

Ранее были рассмотрены лишь основные этапы QR-алгоритма. Его реализация на практике сопровождается еще целым рядом важных деталей. Так, например, для уменьшения относительной погрешности результатов исходная матрица подвергается процедуре *уравновешивания* (или так называемой *балансировки*), уменьшающей норму матрицы $\mathbf{A}$, и т. п.

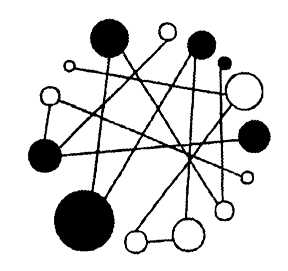
Высокая популярность QR-алгоритма на практике связана не только с его надежностью и высокой вычислительной эффективностью. Важную роль играет тот факт, что подобные преобразования осуществляются с ортогональными матрицами. Предположим, что на шаге с номером $k+1$ $\left(\mathbf{A}_{k+1} = \mathbf{Q}_k^{-1}\mathbf{A}_k\mathbf{Q}_k = \mathbf{Q}_k^T\mathbf{A}_k\mathbf{Q}_k\right)$ элементы матрицы $\mathbf{A}_k$ задавались с погрешностью $\Delta\mathbf{A}_k$. Тогда, оценивая погрешность $\Delta\mathbf{A}_{k+1}$ по норме, получаем:

$$\mathbf{A}_{k+1} + \Delta\mathbf{A}_{k+1} = \mathbf{Q}_k^T\left(\mathbf{A}_k + \Delta\mathbf{A}_k\right)\mathbf{Q}_k = \mathbf{A}_{k+1} + \mathbf{Q}_k^T\Delta\mathbf{A}_k\mathbf{Q}_k,$$

$$\Delta\mathbf{A}_{k+1} = \mathbf{Q}_k^{\mathrm{T}}\Delta\mathbf{A}_k\mathbf{Q}_k, \quad \left\|\Delta\mathbf{A}_{k+1}\right\| \le \left\|\mathbf{Q}_k\right\| \cdot \left\|\mathbf{Q}_k^{\mathrm{T}}\right\| \cdot \left\|\Delta\mathbf{A}_k\right\| = \mathrm{cond}(\mathbf{Q}_k)\left\|\Delta\mathbf{A}_k\right\|.$$

Если в качестве нормы выбрана спектральная норма, то для ортогональной матрицы $\mathbf{Q}_k$ число обусловленности $\mathrm{cond}(\mathbf{Q}_k) = 1$, что и определяет минимальное влияние погрешности предыдущего шага.

Глава 3



Решение нелинейных уравнений и систем

Большинство методов, предназначенных для нахождения корней нелинейного уравнения

$$f(x)=0 \qquad (3.0.1)$$

предполагает, что заранее определены некоторые промежутки, где это уравнение имеет только один корень. Поэтому задаче нахождения решения (3.0.1) с заданной точностью предшествует этап *отделения корней*, связанный с исследованием количества, характера расположения корней и нахождением их грубого приближения. Уравнение (3.0.1) может вообще не иметь решений, а может встретиться ситуация (например, уравнение $x=tg(x)$), когда число корней бесконечно. Этот этап формализуется лишь частично и чаще относится к области математического искусства. Универсального эффективного метода в общем случае нет. На практике в большинстве случаев ограничиваются приближенным построением графика $y=f(x)$ или составлением таблицы для $f(x)$ с некоторым шагом и нахождением участков, где функция меняет знак. При этом шаг не должен быть слишком крупным (должна быть уверенность в не более, чем одном нуле между узлами), а с другой стороны, он не должен быть излишне мал (иначе резко возрастает объем вычислений).

Иногда уравнение (3.0.1) удобно преобразовать к виду

$$\varphi(x)=\mu(x),$$

а затем искать точку пересечения графиков $y=\varphi(x)$ и $y=\mu(x)$, что и будет начальным приближением.

Из нелинейных уравнений общего вида (3.0.1) часто выделяют алгебраические, специфические свойства которых можно использовать при решении.

3.1. Уточнение корней одного уравнения

В рамках раздела будем использовать следующие обозначения: $x^* = x_n + \varepsilon_n$, где x^* — точное решение (3.0.1), x_n — очередное приближение к решению, ε_n — погрешность.

Остановимся лишь на вещественных корнях уравнения (3.0.1), считая, что функция $f(x)$ нужное число раз непрерывно дифференцируема для выбранного метода и установлен промежуток $[a, b]$, где находится единственный корень. При этом $f(a) \cdot f(b) < 0$. Тогда наиболее простым и абсолютно надежным способом является *метод бисекции* (или *метод дихотомии*, или *метод половинного деления*). Его алгоритм представим следующим образом:

1. Вычислить $f(a)$ и $f(b)$.
2. Положить $c = (a+b)/2$ и вычислить $f(c)$.
3. Если $\operatorname{sign}(f(a)) = \operatorname{sign}(f(c))$, заменить a на c. Иначе заменить b на c.
4. Если $|b-a| > \varepsilon$, то перейти к шагу 2. Иначе закончить вычисления.

Одна итерация алгоритма позволяет *гарантированно* сократить исходный промежуток в два раза *независимо от вида функции*.

Вопрос 13

Это свойство метода является его достоинством или недостатком?

Например, для уточнения трех десятичных цифр требуется 10 итераций ($2^{10} = 1024 \approx 10^3$).

Количество чисел, представимых в компьютере, конечно, и в нем может не существовать числа, обеспечивающего строго точное равенство в уравнении (3.0.1). Поэтому минимальная длина промежутка, где $f(x)$ меняет знак, ограничена снизу дискретностью представления чисел в компьютере.

Другой алгоритм, называемый *методом секущих* (рис. 3.1), можно построить, используя интерполяционный полином Лагранжа первой степени для $f(x)$ по двум узлам a и b.

Тогда нуль этого полинома принимается в качестве очередного приближения к корню уравнения (3.0.1).

$$Q_1(x)=\frac{x-b}{a-b}f(a)+\frac{x-a}{b-a}f(b);\ c=a-\frac{b-a}{f(b)-f(a)}f(a).$$

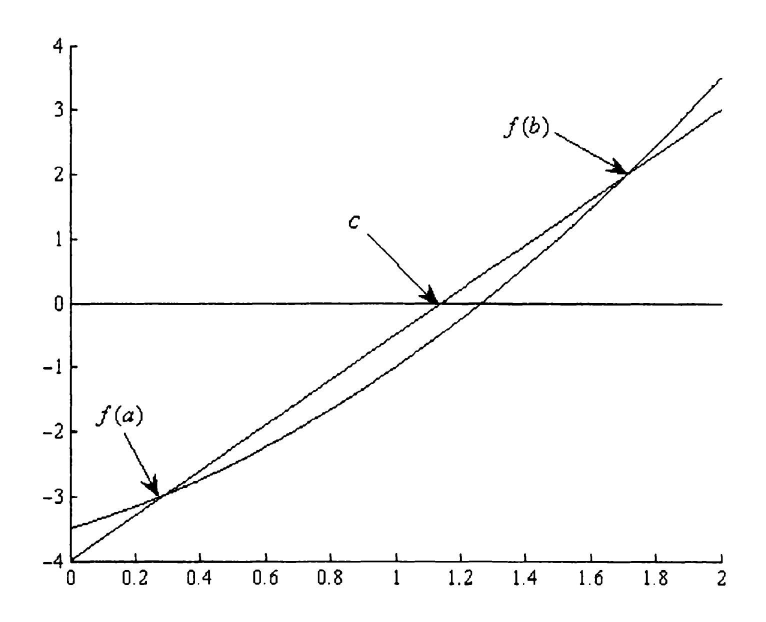


Рис. 3.1. Метод секущих

Новый промежуток будет $[c,\ b]$ или $[a,\ c]$ в зависимости от знака $f(x)$ в точке c. На рис. 3.1 это будет $[c, b]$. Скорость сходимости метода секущих определяется неравенством $\left|x^* - x_{k+1}\right| \le \left|x^* - x_k\right|^{1.618}$. Следует отметить, что замедление сходимости этого алгоритма часто наблюдается, когда очередное приближение получается слишком близко к одному из концов промежутка.

Если функция вычислена более, чем в двух точках, то эта информация может быть использована в дальнейшем. Так, в *методе обратной квадратичной интерполяции* строится интерполяционный полином второй степени по трем точкам x_k, x_{k-1}, x_{k-2} для обратной функции с выполнением условий $x_i = g(f_i)$, $i=k,\ k-1,\ k-2$. В качестве следующего приближения берется $x_{k+1} = g(0)$. Одна из предыдущих точек удаляется. Важно, чтобы три значе-

ния f_i были бы различными, тогда исключается деление на ноль, и сходимость метода определяется неравенством $\left|x^* - x_{k+1}\right| \le \left|x^* - x_k\right|^{1.839}$.

Сочетание методов бисекции и обратной квадратичной интерполяции реализовано в процедуре-функции ZEROIN(A, B, F, EPS) [14]. Здесь A и B — концы интервала, где ищется корень; F — имя процедуры-функции, имеющей лишь один аргумент, для которого вычисляется $f(x)$, EPS — граница погрешности, допустимой в результате.

Основным алгоритмом является метод обратной квадратичной интерполяции (если x_k, x_{k-1}, x_{k-2} не являются различными, то используется метод секущих). Если очередное приближение получается слишком близким к одному из краев промежутка, то осуществляется переключение на метод бисекции.

Еще одним методом для решения (3.0.1) является *метод простой итерации*, для построения которого эквивалентными преобразованиями приведем (3.0.1) к виду

$$x = \varphi(x), \tag{3.1.1}$$

где корень $\mathbf{x}^*$ уравнения (3.1.1) является корнем и (3.0.1). Вместо уравнения (3.1.1) предлагается решать разностное уравнение

$$x_{n+1} = \varphi(x_n) \tag{3.1.2}$$

пошаговым методом. Для оценки сходимости запишем равенство (3.1.1) в точке $\mathbf{x}^*$ и вычтем из него равенство (3.1.2):

$$\varepsilon_{n+1} = x^* - x_{n+1} = \varphi(x^*) - \varphi(x_n) = \varphi(x_n + \varepsilon_n) - \varphi(x_n).$$

Раскладывая $\varphi(x_n + \varepsilon_n)$ в ряд по степеням ε_n и ограничиваясь в остаточном члене первой производной, получаем уравнение погрешности:

$$\varepsilon_{n+1} = \varphi(x_n + \varepsilon_n) - \varphi(x_n) = \varphi(x_n) + \varepsilon_n\varphi'(\eta) - \varphi(x_n) = \varepsilon_n\varphi'(\eta).$$

Отсюда непосредственно следует, что для убывания погрешности необходимо потребовать выполнение условия

$$\left|\varphi'(\eta)\right| < 1. \tag{3.1.3}$$

Искусство пользователя, таким образом, заключается в приведении уравнения (3.0.1) к виду (3.1.1) так, чтобы имело место неравенство (3.1.3). При этом чем меньше по модулю значение производной, тем быстрее достигается желаемая точность.

Высокой скоростью сходимости в ряде случаев обладает *метод Ньютона* (или *метод касательных*). Подставляя в уравнение (3.0.1) его корень $\mathbf{x}^*$ и раскладывая в ряд по степеням ε_n

$$0 = f(x^*) = f(x_n + \varepsilon_n) = f(x_n) + \varepsilon_n f'(x_n) + \frac{\varepsilon_n^2}{2!} f''(\eta), \tag{3.1.4}$$

пренебрежем последним слагаемым в (3.1.4) и для ε_n получим

$$\varepsilon_n \approx -\frac{f(x_n)}{f'(x_n)}.$$

Тогда рабочая формула метода Ньютона приобретает вид:

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}. \tag{3.1.5}$$

В отличие от методов бисекции, секущих и обратной квадратичной интерполяции сходимость (3.1.5) обеспечивается далеко не всегда. В перечне достаточных условий сходимости фигурирует не только существование ненулевой производной $f'(x)$ в точках x_n, но и ее знакопостоянство. Если, тем не менее, методу Ньютона обеспечено хорошее начальное приближение, то в дальнейшем убывание погрешности носит квадратичный характер. Для доказательства этого факта из очевидного равенства $x^* = x^*$ вычтем уравнение (3.1.5):

$$\varepsilon_{n+1} = \varepsilon_n + \frac{f(x_n)}{f'(x_n)} = \frac{f(x_n) + \varepsilon_n f'(x_n)}{f'(x_n)}. \tag{3.1.6}$$

Упрощая числитель (3.1.6) с помощью равенства (3.1.4), получаем

$$\varepsilon_{n+1} = -\frac{f''(\eta)}{2f'(x_n)}\varepsilon_n^2,$$

$$|\varepsilon_{n+1}| < C\varepsilon_n^2, \quad \left|\frac{f''(\eta)}{2f'(x_n)}\right| < C. \tag{3.1.7}$$

Метод Ньютона называют еще *методом касательных*, т. к. формула (3.1.5) получается на основе уравнения касательной

$$y = Q_1(x) = f(x_n) + f'(x_n)(x - x_n).$$

Корень этого полинома отвечает следующему приближению x_{n+1}.

Для расширения области сходимости можно использовать метод Ньютона с регулировкой шага:

$$x_{n+1} = x_n - \alpha_n \frac{f(x_n)}{f'(x_n)}, \quad 0 < \alpha_n \le 1. \tag{3.1.8}$$

Первоначально, когда начальное приближение x_0 еще далеко от x^*, параметр α_n выбирают меньше 1 (часто на практике это примерно 1/3), а по мере приближения x_n к x^* значение $\alpha_n \to 1$, превращая (3.1.8) в обычный метод Ньютона. В некоторых случаях это позволяет расширить область сходимости метода Ньютона.

Широкое распространение получил и модифицированный метод Ньютона с постоянным значением производной:

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_0)}. \tag{3.1.9}$$

Сходимость в этом случае несколько замедляется, но заметно уменьшается трудоемкость отдельной итерации, не требующей теперь вычисления производной.

3.2. Метод Ньютона для систем уравнений

Решение систем нелинейных уравнений доставляет очень большие трудности, т. к. нет универсальных алгоритмов решения этих задач, особенно для больших m.

Достоинством метода Ньютона по сравнению со многими алгоритмами предыдущего раздела является то, что он обобщается на системы уравнений. С этой целью обратимся к уравнению (3.0.1), полагая $\mathbf{x}$ и $\mathbf{f}$ векторами $\mathbf{x} = \left(x^{(1)}, x^{(2)}, \ldots, x^{(m)}\right)^T$, $\mathbf{f} = \left(f^{(1)}, f^{(2)}, \ldots, f^{(m)}\right)^T$.

Формула, являющаяся аналогом (3.1.5), может быть получена таким же образом, как и для скалярного случая, на основе разложения типа (3.1.4). При этом ε_n представляет собой вектор, и разложение в ряд необходимо проводить по всем компонентам этого вектора для $\mathbf{f}$, как функции многих переменных. В настоящем разделе используем другой подход, основанный на лемме Адамара.

Рассмотрим вектор-функцию $\mathbf{f}(\mathbf{x}+t\cdot\varepsilon_n)$ скалярного аргумента t. При этом величина ε_n считается не зависящей от t. Вычислим производную этой функции по t и проинтегрируем результат по t от 0 до 1.

$$\frac{d}{dt}\mathbf{f}(\mathbf{x}+t\cdot\varepsilon_n)=\frac{\partial\mathbf{f}(\mathbf{x}+t\cdot\varepsilon_n)}{\partial(\mathbf{x}+t\cdot\varepsilon_n)}\cdot\frac{d(\mathbf{x}+t\cdot\varepsilon_n)}{dt}=\frac{\partial\mathbf{f}(\mathbf{x}+t\cdot\varepsilon_n)}{\partial(\mathbf{x}+t\cdot\varepsilon_n)}\varepsilon_n;$$

$$\int_0^1\frac{d}{dt}\mathbf{f}(\mathbf{x}+t\cdot\varepsilon_n)dt=\int_0^1\frac{\partial\mathbf{f}(\mathbf{x}+t\cdot\varepsilon_n)}{\partial\mathbf{x}}dt\cdot\varepsilon_n;$$

$$\mathbf{f}(\mathbf{x}+\varepsilon_n)-\mathbf{f}(\mathbf{x})=\int_0^1\frac{\partial\mathbf{f}(\mathbf{x}+t\cdot\varepsilon_n)}{\partial\mathbf{x}}dt\cdot\varepsilon_n. \tag{3.2.1}$$

Полагая $\mathbf{x}=\mathbf{x}_n$ и по-прежнему считая величину $\mathbf{x}^*=\mathbf{x}_n+\varepsilon_n$ точным корнем уравнения (3.0.1), т. е. $\mathbf{f}(\mathbf{x}_n+\varepsilon_n)=\mathbf{f}(\mathbf{x}^*)=0$, из (3.2.1) имеем:

$$\varepsilon_n=-\left(\int_0^1\frac{\partial\mathbf{f}(\mathbf{x}+t\cdot\varepsilon_n)}{\partial\mathbf{x}}dt\right)^{-1}\mathbf{f}(\mathbf{x}_n).$$

Вычисляя интеграл в последней формуле по квадратурной формуле левых прямоугольников, получаем метод Ньютона для систем уравнений в традиционной форме

$$\mathbf{x}_{n+1}=\mathbf{x}_n-\left(\frac{\partial\mathbf{f}(\mathbf{x}_n)}{\partial\mathbf{x}}\right)^{-1}\mathbf{f}(\mathbf{x}_n) \tag{3.2.2}$$

или

$$\frac{\partial\mathbf{f}(\mathbf{x}_n)}{\partial\mathbf{x}}(\mathbf{x}_{n+1}-\mathbf{x}_n)=-f(\mathbf{x}_n). \tag{3.2.3}$$

Представление метода в виде (3.2.3) позволяет уменьшить вычислительные затраты, поскольку не требует обращения матрицы Якоби $\frac{\partial\mathbf{f}(\mathbf{x}_n)}{\partial\mathbf{x}}$, а сводится к решению линейной алгебраической системы на каждом шаге итерационного процесса. Как и в скалярном случае, в достаточно малой окрестности корня итерации сходятся и скорость сходимости квадратичная. Значительно уменьшается объем вычислений в модифицированном варианте метода Ньютона (формула (3.1.9) является его скалярным вариантом), когда матрица Якоби вычисляется однократно, раскладывается в произведение треугольных

матриц программой DECOMP, а затем для получения очередного приближения используется только программа SOLVE.

Для решения систем уравнений может быть использован и метод простой итерации. Формула (3.1.2) сохраняет прежний вид, только $\mathbf{x}$ и $\varphi(\mathbf{x})$ являются векторами. Достаточным условием сходимости является выполнение неравенства $\left\|\frac{\partial\varphi}{\partial\mathbf{x}}\right\| < 1$, где $\frac{\partial\varphi}{\partial\mathbf{x}}$ — матрица Якоби.

Подведем некоторые итоги. Для скалярного уравнения (3.0.1) только в рамках *разд. 3.1* были рассмотрены пять методов, каждый из которых имеет свои сильные стороны. Методы бисекции, секущих, обратной квадратичной интерполяции весьма надежны и абсолютно застрахованы от неудачи. Метод Ньютона в окрестности нуля демонстрирует квадратичную скорость сходимости. Таким образом, пользователю предлагается широкий выбор алгоритмов, и он может ориентироваться на специфику решаемой задачи. Иначе обстоит дело при решении систем уравнений (3.0.1), когда $\mathbf{f}$ и $\mathbf{x}$ — векторы. Простейшие методы из перечисленных на системы уравнений впрямую не могут быть распространены, метод Ньютона требует хорошего начального приближения (а как его обеспечить?), а метод простой итерации требует предварительного преобразования системы (3.0.1) к виду (3.1.1) с выполнением условия $\left\|\frac{\partial\varphi}{\partial\mathbf{x}}\right\| < 1$, что в общем случае весьма не тривиально. Все сказанное требует серьезного усиления позиций алгоритмов для систем уравнений, в первую очередь, в направлении надежного получения решения. К этим проблемам мы вернемся в *главе 6*.

3.3. Методы минимальных невязок Ракитского

Решение систем нелинейных уравнений может быть построено на идее минимизации некоторой функции подобно тому, как для линейных систем строился метод минимальных невязок (2.3.16)—(2.3.17). Так, для (3.0.1) Ю. В. Ракитским были предложены однопараметрический и трехпараметрический методы минимальных невязок. Начнем с однопараметрического метода, полагая, что $\mathbf{f}(\mathbf{x})$ — дважды дифференцируемая вектор-функция векторного аргумента, отделение корней произведено, и начальное приближение $\mathbf{x}_0$ находится в некоторой r-окрестности корня: $\|\mathbf{x} - \mathbf{x}_0\| < r$.

На первой итерации новое приближение находится по формуле

$$\mathbf{x}_1 = \mathbf{x}_0 + \alpha_0 \mathbf{f}(\mathbf{x}_0).$$

Учитывая (3.0.1), построим невязку в виде $R(\alpha_0) = (\mathbf{f}(\mathbf{x}_1), \mathbf{f}(\mathbf{x}_1))$ и минимизируем ее по параметру α_0. При достаточно малых значениях $|\alpha_0| < \delta$ разложим $R(\alpha_0)$ в степенной ряд по степеням α_0:

$$R(\alpha_0) = R(0) + \alpha_0 R'(0) + \frac{\alpha_0^2}{2!} R''(0) + O(\alpha_0^3). \tag{3.3.1}$$

Минимум обеспечивается условием $R'(\alpha_0) = 0$, которое при пренебрежении в (3.3.1) слагаемыми с α_0 выше второй степени приводит к следующему требованию на α_0:

$$\alpha_0 = -\frac{R'(0)}{R''(0)}. \tag{3.3.2}$$

Если в частном случае система (3.0.1) является линейной, то $\mathbf{f}(\mathbf{x})$ является вектором невязки, и формула (3.3.2) переходит в выражение (2.3.17) метода минимальных невязок для систем (2.0.1). В нелинейном же случае значения $R'(0)$ и $R''(0)$ вычисляются по формулам численного дифференцирования. Выберем некоторое значение $|h_0| < \delta$ и введем обозначения: $R_0 = R(0)$, $R_+ = R(h_0)$, $R_- = R(-h_0)$. Тогда

$$R'(0) \approx \frac{R_+ - R_-}{2h_0}, \quad R''(0) \approx \frac{R_+ - 2R_0 + R_-}{h_0^2},$$

и формула для α_0 приобретает вид

$$\alpha_0 = -\frac{R'(0)}{R''(0)} \approx -\frac{h_0}{2} \cdot \frac{R_+ - R_-}{R_+ - 2R_0 + R_-}.$$

Последующие итерации выполняются аналогично:

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{f}(\mathbf{x}_k), \tag{3.3.3}$$

$$\alpha_k = -\frac{R'(0)}{R''(0)} \approx -\frac{h_k}{2} \cdot \frac{R(h_k) - R(-h_k)}{R(h_k) - 2R(0) + R(-h_k)}. \tag{3.3.4}$$

На каждом шаге вектор-функция $\mathbf{f}(\mathbf{x})$ вычисляется три раза, а условие успешного выполнения итераций выглядит следующим образом:

$$\|\mathbf{f}(\mathbf{x}_k)\| < \|\mathbf{f}(\mathbf{x}_{k-1})\|.$$

Теперь обратимся к трехпараметрическому методу. По формулам однопараметрического метода для выбранного начального приближения $\mathbf{x}_0$ находим

$$\mathbf{x}_1^{(0)} = \mathbf{x}_0 + \alpha_0 \mathbf{f}(\mathbf{x}_0).$$

Вычисляем α_0, минимизируя невязку $R(\alpha_0) = \left(\mathbf{f}(\mathbf{x}_1^{(0)}), \mathbf{f}(\mathbf{x}_1^{(0)})\right)$.

Далее вычисляем $\mathbf{f}(\mathbf{x}_1^{(0)})$ и строим вектор $\mathbf{P}_0 = \mathbf{f}(\mathbf{x}_0) + \beta_0 \mathbf{f}\left(\mathbf{x}_1^{(0)}\right)$, зависящий от параметра β_0, значение которого находим из условия ортогональности $\mathbf{P}_0$ и $\mathbf{f}(\mathbf{x}_0)$:

$$(\mathbf{P}_0, \mathbf{f}(\mathbf{x}_0)) = \left(\mathbf{f}(\mathbf{x}_0) + \beta_0 \mathbf{f}\left(\mathbf{x}_1^{(0)}\right), \mathbf{f}(\mathbf{x}_0)\right) = 0;$$

$$\beta_0 = -\frac{(\mathbf{f}(\mathbf{x}_0), \mathbf{f}(\mathbf{x}_0))}{\left(\mathbf{f}(\mathbf{x}_0), \mathbf{f}\left(\mathbf{x}_1^{(0)}\right)\right)}.$$

Затем строим вектор $\mathbf{x}_1$ по формуле $\mathbf{x}_1 = \mathbf{x}_0 + \alpha_0 \mathbf{f}(\mathbf{x}_0) + \gamma_0 \mathbf{P}_0$, где параметр γ_0 находим из условия минимума скалярного произведения:

$$S(\gamma_0) = (\mathbf{f}(\mathbf{x}_1), \mathbf{f}(\mathbf{x}_1)).$$

Как и в однопараметрическом методе, раскладываем $S(\gamma_0)$ в ряд и по условию минимальности приходим к равенству $\gamma_0 = -\dfrac{S'(0)}{S''(0)}$. Аналогично воспользуемся формулами численного дифференцирования с шагом Δ_0: $(|\Delta_0| < \delta)$:

$$\gamma_0 = -\frac{S'(0)}{S''(0)} \approx -\frac{\Delta_0}{2} \cdot \frac{S(\Delta_0) - S(-\Delta_0)}{S(\Delta_0) - 2S(0) + S(-\Delta_0)}.$$

Последующие итерации выполняются в соответствии с равенствами

$$\mathbf{x}_{k+1}^{(0)} = \mathbf{x}_k + \alpha_k \mathbf{f}(\mathbf{x}_k), \quad R(\alpha_k) = \left(\mathbf{f}(\mathbf{x}_{k+1}^{(0)}), \mathbf{f}(\mathbf{x}_{k+1}^{(0)})\right),$$

$$\alpha_k \approx -\frac{h_k}{2} \cdot \frac{R(h_k) - R(-h_k)}{R(h_k) - 2R(0) + R(-h_k)}.$$

Так завершается прогноз по однопараметрическому методу. Далее вычисления проводятся по формулам:

$$\mathbf{P}_k = \mathbf{f}\left(\mathbf{x}_k\right) + \beta_k \mathbf{f}\left(\mathbf{x}_{k+1}^{(0)}\right); \quad \beta_k = -\frac{\left(\mathbf{f}(\mathbf{x}_k), \mathbf{f}(\mathbf{x}_k)\right)}{\left(\mathbf{f}(\mathbf{x}_k), \mathbf{f}\left(\mathbf{x}_{k+1}^{(0)}\right)\right)};$$

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{f}(\mathbf{x}_k) + \gamma_k \mathbf{P}_k;$$

$$S(\gamma_k) = \left(\mathbf{f}(\mathbf{x}_{k+1}), \mathbf{f}(\mathbf{x}_{k+1})\right); \quad \gamma_k \approx -\frac{\Delta_k}{2} \cdot \frac{S(\Delta_k) - S(-\Delta_k)}{S(\Delta_k) - 2S(0) + S(-\Delta_k)}.$$

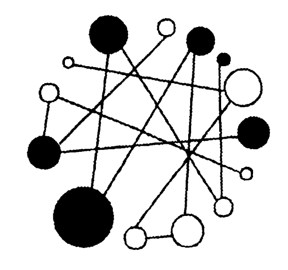
В трехпараметрическом методе шесть вычислений значений вектор-функции $\mathbf{f}(\mathbf{x})$ на каждой итерации:

$$\mathbf{f}(\mathbf{x}_k), \quad \mathbf{f}(\mathbf{x}_k + h_k \mathbf{f}(\mathbf{x}_k)), \quad \mathbf{f}(\mathbf{x}_k - h_k \mathbf{f}(\mathbf{x}_k)),$$

$$\mathbf{f}(\mathbf{x}_{k+1}^{(0)}), \quad \mathbf{f}(\mathbf{x}_{k+1}^{(0)} + \Delta_k \mathbf{f}(\mathbf{x}_k)), \quad \mathbf{f}(\mathbf{x}_{k+1}^{(0)} - \Delta_k \mathbf{f}(\mathbf{x}_k)).$$

Условия успешного выполнения итераций: $\left\|\mathbf{f}\left(\mathbf{x}_{k+1}\right)\right\| \le \left\|\mathbf{f}\left(\mathbf{x}_{k+1}^{0}\right)\right\| \le \left\|\mathbf{f}\left(\mathbf{x}_k\right)\right\|$. Вопросы сходимости этих методов еще не в полной мере изучены.

Глава 4



Решение дифференциальных уравнений

Как известно, в практических приложениях решения дифференциального уравнения или системы уравнений описывают динамику разнообразных явлений и процессов (например, движение совокупности взаимодействующих материальных точек, химическую кинетику, процессы в электрических цепях и т. п.). Однако интегрируемых в явном виде дифференциальных уравнений чрезвычайно мало. Поэтому столь важны численные методы.

Задача Коши из множества решений для системы

$$\frac{d\mathbf{x}(t)}{dt} = \mathbf{f}\left(t, \mathbf{x}(t)\right), \tag{4.0.1}$$

где t — независимая переменная, $\mathbf{x}(t) = \left(x^{(1)}, ..., x^{(m)}\right)^{\mathrm{T}}$ — вектор искомых функций, удовлетворяющих уравнению, $\mathbf{f}(t, \mathbf{x})$ — вектор заданных, нужное число раз дифференцируемых функций, выделяет одно, проходящее через начальную точку $(t_0, \mathbf{x}_0)$. Аналогично ставится задача и для дифференциального уравнения m-го порядка, разрешенного относительно старшей производной, которое сводится к системе (4.0.1) из m уравнений первого порядка.

Если правые части $\mathbf{f}(t, \mathbf{x})$, а также элементы матрицы Якоби $\dfrac{\partial \mathbf{f}}{\partial \mathbf{x}}$ непрерывны и ограничены в некоторой окрестности точки $(t_0, \mathbf{x}_0)$, то задача Коши имеет единственное решение. Первоначально, исключительно для простоты рассуждений, будем полагать, что (4.0.1) представляет собой одно уравнение. Вместе с тем, абсолютно все излагаемые в данной главе методы сохраняют

свой внешний вид и для случая, когда **x** и **f** являются векторами и (4.0.1) является системой уравнений.

Численное решение, получаемое любым способом, характеризуется двумя важными свойствами: устойчивостью и точностью. Определение устойчивости, данное А. М. Ляпуновым *(подробнее см. разд. П3.9)*, связано с качественным характером изменения решения при внесении в него возмущений. Точность же характеризует отличие приближенного решения от точного (если есть возможность получить или оценить последнее).

Общий подход к решению (4.0.1) заключается в приближенном сведении дифференциального уравнения к некоторому разностному уравнению, которое, в свою очередь, решается затем пошаговым методом. С этой целью выполним дискретизацию независимой переменной: $t_n = t_0 + nh$, где h — шаг интегрирования (шаг дискретности), а значения решения и его производной в этих точках кратко обозначим как $\mathbf{x}_n = \mathbf{x}(t_n)$ и $\mathbf{f}_n = \mathbf{f}(t_n, \mathbf{x}_n)$. Интегрируя (5.0.1) на промежутке $[t_n, t_{n+1}]$, получаем формулу

$$\mathbf{x}_{n+1} = \mathbf{x}_n + \int_{t_n}^{t_{n+1}} \mathbf{f}(\tau, \mathbf{x}(\tau))d\tau, \tag{4.0.2}$$

которую можно считать базовой для построения большей части известных разностных схем. Различные методы при этом отличаются способом вычисления интеграла в равенстве (4.0.2).

Использование квадратурных формул левых и правых прямоугольников, а также формулы трапеций, приводит, соответственно, к следующим численным методам:

$$\mathbf{x}_{n+1} = \mathbf{x}_n + h\mathbf{f}(t_n, \mathbf{x}_n), \tag{4.0.3}$$

$$\mathbf{x}_{n+1} = \mathbf{x}_n + h\mathbf{f}(t_{n+1}, \mathbf{x}_{n+1}), \tag{4.0.4}$$

$$\mathbf{x}_{n+1} = \mathbf{x}_n + \frac{h}{2}\left(\mathbf{f}(t_n, \mathbf{x}_n) + \mathbf{f}(t_{n+1}, \mathbf{x}_{n+1})\right), \tag{4.0.5}$$

которые получили название *явного метода ломаных Эйлера*, *неявного метода ломаных Эйлера* и *неявного метода трапеций*. Разностные уравнения (4.0.4) и (4.0.5) неявно задают значения $\mathbf{x}_{n+1}$ и требуют решения нелинейных уравнений на каждом шаге интегрирования.

Каковы дальнейшие пути повышения точности? На первый взгляд, возможно использование более точных квадратурных формул, например, формулы Симпсона, других формул Ньютона — Котеса, Чебышева, Гаусса и др. Одна-

ко все они требуют вычисления подынтегральной функции в некоторых точках внутри промежутка интегрирования, в то время как решение $\mathbf{x}(t)$ в этих точках нами не определено. Поэтому для решения задачи используются другие подходы.

4.1. Методы Адамса. Локальная и глобальная погрешности. Степень метода

Первый подход предполагает для построения решения в точке t_{n+1} использование информации в ранее полученных точках t_n, t_{n-1}, ... Так, по двум предыдущим точкам t_n и t_{n-1} построим интерполяционный полином первой степени для функции $\mathbf{f}(t,\mathbf{x})$

$$\mathbf{f}\left(\tau,\mathbf{x}(\tau)\right)\approx\frac{\tau-t_{n-1}}{t_n-t_{n-1}}\mathbf{f}_n+\frac{\tau-t_n}{t_{n-1}-t_n}\mathbf{f}_{n+1}$$

и подставим его в формулу (4.0.2). Попутно заметим, что полином используется вне промежутка интерполирования, т. е. проводится экстраполяция. Получаем следующий численный метод:

$$\mathbf{x}_{n+1}=\mathbf{x}_n+\frac{h}{2}\left(3\mathbf{f}_n-\mathbf{f}_{n-1}\right). \tag{4.1.1}$$

Использование трех точек t_n, t_{n-1}, t_{n-2} и полинома второй степени приведет к формуле

$$\mathbf{x}_{n+1}=\mathbf{x}_n+\frac{h}{12}\left(23\mathbf{f}_n-16\mathbf{f}_{n-1}+5\mathbf{f}_{n-2}\right), \tag{4.1.2}$$

а для четырех точек разностная схема алгоритма приобретает вид

$$\mathbf{x}_{n+1}=\mathbf{x}_n+\frac{h}{24}\left(55\mathbf{f}_n-59\mathbf{f}_{n-1}+37\mathbf{f}_{n-2}-9\mathbf{f}_{n-3}\right). \tag{4.1.3}$$

Все эти методы получили название *методов Адамса*. Они принадлежат семейству многошаговых алгоритмов, разностные уравнения которых имеют порядок выше первого. Методы (4.1.1)—(4.1.3) являются *явными* методами Адамса. Если в состав точек, по которым строится интерполяционный поли-

ном, включить t_{n+1}, то возникают *неявные* методы Адамса. Для двух точек t_{n+1}, t_n получается метод трапеций (4.0.5), а для трех точек t_{n+1}, t_n, t_{n-1} и четырех точек t_{n+1}, t_n, t_{n-1}, t_{n-2} — следующие два метода:

$$\mathbf{x}_{n+1} = \mathbf{x}_n + \frac{h}{12}\left(5\mathbf{f}_{n+1} + 8\mathbf{f}_n - \mathbf{f}_{n-1}\right), \qquad (4.1.4)$$

$$\mathbf{x}_{n+1} = \mathbf{x}_n + \frac{h}{24}\left(9\mathbf{f}_{n+1} + 19\mathbf{f}_n - 5\mathbf{f}_{n-1} + \mathbf{f}_{n-2}\right). \qquad (4.1.5)$$

До изучения дальнейшего материала полезно попытаться ответить на следующий вопрос.

Вопрос 14

Если в любой из формул (4.1.1)—(4.1.5) сложить все коэффициенты, стоящие при значениях $\mathbf{f}(t,\mathbf{x})$ в различных точках, то получается единица. Случайно ли это?

Несомненным достоинством явных методов Адамса является тот факт, что все они независимо от своей точности требуют лишь однократного вычисления функции $\mathbf{f}(t,\mathbf{x})$ на одном шаге, и конкурировать с ними в этом плане весьма трудно. Остальные значения производной решения берутся с предыдущих шагов. Вместе с тем, методы Адамса, как и другие многошаговые алгоритмы, не являются самостартующими, т. е. они требуют для начала интегрирования специальных стартовых алгоритмов для расчета дополнительных начальных условий. В качестве этих стартовых методов может быть использован любой другой метод, например метод Рунге — Кутты, рассматриваемый далее. Специфический способ представляют стартовые алгоритмы Ракитского, имеющие асимптотический характер. В частности, для приведенных выше методов Адамса (вплоть до четвертой степени) достаточно вычислять начальные точки по формуле:

$$\mathbf{x}_{-k} = \mathbf{x}_0 - kh\mathbf{f}_0, \quad k = 1, 2, 3;$$

причем уже после нескольких первых шагов обеспечивается точность, адекватная степени выбранного метода.

Неявные методы Адамса могут использоваться как сами по себе (тогда на каждом шаге решаются нелинейные уравнения относительно $\mathbf{x}_{n+1}$), так и в

паре с явными методами. В последнем случае значение $\mathbf{x}_{n+1}$ сначала оценивается явным методом $\left(\mathbf{x}_{n+1}^{\text{э}}\right)$, а затем уточняется неявным алгоритмом. Например, такую пару методов образуют методы (4.1.3) и (4.1.5)

$$\mathbf{x}_{n+1}^{\text{э}} = \mathbf{x}_n + \frac{h}{24}\left(55\mathbf{f}_n - 59\mathbf{f}_{n-1} + 37\mathbf{f}_{n-2} - 9\mathbf{f}_{n-3}\right), \qquad (4.1.6)$$

$$\mathbf{x}_{n+1} = \mathbf{x}_n + \frac{h}{24}\left(9\mathbf{f}\left(t_{n+1}, \mathbf{x}_{n+1}^{\text{э}}\right) + 19\mathbf{f}_n - 5\mathbf{f}_{n-1} + \mathbf{f}_{n-2}\right). \qquad (4.1.7)$$

В зарубежной литературе совместное использование явного и неявного методов называют *методами прогноза — коррекции*. В нашей литературе часто используют термин *экстраполяционные* методы для (4.1.6) и *интерполяционные* методы для (4.1.7). Аналогичные пары методов образуют (4.1.1) и (4.0.5), (4.1.2) и (4.1.4).

Теперь обратимся к анализу погрешности численных методов и начнем с самого простого алгоритма — явного метода ломаных Эйлера. Рассмотрим частный случай формулы (4.0.3), когда функция $\mathbf{f}(t,\mathbf{x})$ не зависит от $\mathbf{x}$. Тогда явный метод ломаных Эйлера превращается в квадратурную формулу левых прямоугольников:

$$\mathbf{x}_{n+1} = \mathbf{x}_n + h\mathbf{f}\left(t_n\right) = \mathbf{x}_0 + h\sum_{k=0}^{n}\mathbf{f}_k. \qquad (4.1.8)$$

При этом общая погрешность в точке t_n является точной суммой погрешностей, допущенных на каждом отдельном шаге.

Иная ситуация складывается, когда $\mathbf{f}(t,\mathbf{x})$ зависит от $\mathbf{x}$. Только погрешность первого шага формулы (4.0.3) при $n=0$ вычисляется аналогично (4.1.8). Уже на втором шаге при $n=1$ эта погрешность сложным образом зависит от погрешности первого шага, т. к. при вычислении $\mathbf{f}(t_1,\mathbf{x}_1)$ используется приближенное значение $\mathbf{x}_1$. В общем случае на n-м шаге погрешность очень сложно зависит от всех погрешностей, допущенных на предыдущих шагах. Разностное уравнение метода может оказаться неустойчивым, и тогда происходит неприемлемый рост погрешности.

Устойчивость разностной схемы связана с выбранным методом, шагом интегрирования и видом функции $\mathbf{f}(t,\mathbf{x})$. Важно так выбрать сам метод и шаг

5 Зак. 3418

для него, чтобы погрешность решения была бы приемлемой. В соответствии со сказанным вводятся погрешности двух видов:

- *локальная погрешность* — погрешность, допущенная на одном шаге при условии, что решение во всех предыдущих точках вычислено точно;
- *глобальная погрешность* — разность между точным и приближенным решением на n-м шаге.

Именно глобальная погрешность является истинной погрешностью. Локальная погрешность совпадает с ней лишь на первом шаге. Однако в общем случае оценка глобальной погрешности крайне затруднена, а чаще невозможна, и поэтому оценивают локальную погрешность на каждом шаге. Малая величина локальной погрешности вовсе не гарантирует малую величину глобальной, но если есть уверенность, что устойчивость разностного уравнения метода обеспечена, то из малой величины локальной погрешности следует, что глобальная не будет слишком велика. Будем пока полагать, что устойчивость обеспечена, и рассмотрим подробнее характеристики локальной погрешности.

Важной характеристикой является *"степень"* (или *"порядок точности"*) метода. В дальнейшем будем употреблять термин *"степень"*, чтобы не путать эту характеристику с порядком разностного уравнения метода. Все ранее рассмотренные методы могут быть записаны в следующем виде:

$$\mathbf{x}_{n+1} = \mathbf{x}_n + h\mathbf{F}\left(t_n, h, \mathbf{x}_{n+1}, \mathbf{x}_n, \mathbf{x}_{n-1}, \ldots, \mathbf{x}_{n-k}\right). \qquad (4.1.9)$$

Разложим выражение в правой части равенства (4.1.9) в ряд Тейлора по степеням h в точке t_n:

$$\mathbf{x}_{n+1} = \mathbf{x}\left(t_n + h\right) = \mathbf{x}_n + \sum_{k=1}^{\infty} \alpha_k(t_n) h^k \frac{d^k \mathbf{x}(t_n)}{dt^k}. \qquad (4.1.10)$$

С другой стороны, значение $\mathbf{x}_{n+1} = \mathbf{x}\left(t_n + h\right)$ может быть представлено, в свою очередь, точным разложением в ряд

$$\mathbf{x}_{n+1} = \mathbf{x}\left(t_n + h\right) = \mathbf{x}_n + \sum_{k=1}^{\infty} \frac{h^k}{k!} \cdot \frac{d^k \mathbf{x}(t_n)}{dt^k}. \qquad (4.1.11)$$

Метод имеет *степень (порядок точности) s*, если коэффициенты разложения (4.1.10) совпадают с соответствующими коэффициентами (4.1.11) до h^s включительно. В качестве примера определим степень некоторых ранее полученных методов.

Для явного метода ломаных Эйлера, учитывая (4.0.1), имеем

$$\mathbf{x}_{n+1} = \mathbf{x}_n + h\mathbf{f}(t_n, \mathbf{x}_n) = \mathbf{x}_n + h\mathbf{x}'(t_n).$$

Вычитая эту формулу из (4.1.11), видим, что совпадают коэффициенты лишь при h^1, и для локальной погрешности этого метода первой степени справедлива оценка: $\varepsilon_{n+1} = \dfrac{h^2\mathbf{x}''(\eta)}{2}$. Аналогично убеждаемся, что метод Адамса (4.1.1) имеет вторую степень (совпадают члены разложения при h^1 и h^2)

$$\mathbf{x}_{n+1} = \mathbf{x}_n + \frac{h}{2}\big(3\mathbf{x}'(t_n) - \mathbf{x}'(t_n - h)\big) = \mathbf{x}_n + h\mathbf{x}'(t_n) + \frac{h^2}{2}\mathbf{x}''(t_n) - \frac{h^3}{4}\mathbf{x}'''(t_n) + \ldots,$$

а после вычитания этого выражения из (4.1.11) получаем оценку локальной погрешности: $\varepsilon_{n+1} = \dfrac{5h^3\mathbf{x}'''(\eta)}{12}$. Нетрудно убедиться, что неявный метод ломаных Эйлера имеет первую степень, неявный метод трапеций — вторую, методы Адамса (4.1.2) и (4.1.4) — третью, а методы (4.1.3) и (4.1.5) — четвертую степень соответственно. Главный член погрешности метода s-ой степени содержит, как множитель, величину h^{s+1}.

В заключение данного раздела отметим любопытную ситуацию. И явный метод ломаных Эйлера, и неявный являются методами первой степени и имеют локальную погрешность примерно одного порядка. Вместе с тем, явный метод обладает много меньшей трудоемкостью каждого шага и в отличие от неявного метода не требует постоянного разрешения уравнения относительно $\mathbf{x}_{n+1}$. Возникает следующий вопрос.

Вопрос 15

Зачем может потребоваться неявный метод ломаных Эйлера, если его локальная погрешность соизмерима с погрешностью явного метода, а объем вычислений на шаге заметно выше?

Ранее уже отмечался главный недостаток методов Адамса — необходимость в стартовых алгоритмах. Анализ формул (4.1.1)—(4.1.5) показывает, что повышение степени метода дается ценой повышения порядка разностного уравнения. Нельзя ли повышать степень метода так, чтобы алгоритм оставался одношаговым? Ответом на этот вопрос является подход, реализованный в методах типа Рунге — Кутты.

4.2. Методы Рунге — Кутты. Программа *RKF45*

Метод трапеций (4.0.5) является неявным. Что произойдет, если вычислить $\mathbf{x}_{n+1}$ сначала по формуле (4.0.3), а затем уточнить по (4.0.5)?

$$\mathbf{x}_{n+1}^{*} = \mathbf{x}_n + h\mathbf{f}\left(t_n, \mathbf{x}_n\right),$$
$$\mathbf{x}_{n+1} = \mathbf{x}_n + \frac{h}{2}\left(\mathbf{f}\left(t_n, \mathbf{x}_n\right) + \mathbf{f}\left(t_{n+1}, \mathbf{x}_{n+1}^{*}\right)\right). \tag{4.2.1}$$

Полученный одношаговый метод, называемый *методом Эйлера — Коши*, является уже явным. Как будет показано, он имеет вторую степень, которая достигается ценой двух вычислений функции $\mathbf{f}(t, \mathbf{x})$ на каждом шаге.

Приведем еще один пример. Сделаем полшага с помощью явного метода ломаных Эйлера, а затем используем полученное значение в квадратурной формуле средних прямоугольников, примененной к интегралу в (4.0.2):

$$\mathbf{x}_{n+1/2}^{*} = \mathbf{x}_n + \frac{h}{2}\mathbf{f}\left(t_n, \mathbf{x}_n\right),$$
$$\mathbf{x}_{n+1} = \mathbf{x}_n + h\mathbf{f}\left(t_n + \frac{h}{2}, \mathbf{x}_{n+1/2}^{*}\right). \tag{4.2.2}$$

Этот метод, называемый *усовершенствованным методом ломаных Эйлера*, также имеет вторую степень и требует двукратного вычисления $\mathbf{f}(t, \mathbf{x})$. Приведенные примеры укладываются в следующую схему. Вычислим $\mathbf{f}(t, \mathbf{x})$ дважды в некоторых точках и их линейную комбинацию используем для получения $\mathbf{x}_{n+1}$:

$$\mathbf{k}_1 = h\mathbf{f}\left(t_n, \mathbf{x}_n\right),$$
$$\mathbf{k}_2 = h\mathbf{f}\left(t_n + \alpha_2 h, \mathbf{x}_n + \beta_{21}\mathbf{k}_1\right), \tag{4.2.3}$$
$$\mathbf{x}_{n+1} = \mathbf{x}_n + p_1\mathbf{k}_1 + p_2\mathbf{k}_2.$$

Параметры p_1, p_2, α_2, β_{21} будем выбирать так, чтобы разложение формулы метода (4.2.3) в ряд максимальным образом совпадало с разложением точного решения (4.1.11). С этой целью отметим, что согласно (4.0.1)

$$\mathbf{x}' = \mathbf{f}(t, \mathbf{x}), \quad \mathbf{x}'' = \frac{\partial \mathbf{f}}{\partial t} + \frac{\partial \mathbf{f}}{\partial \mathbf{x}}\mathbf{f},$$

и (4.1.11) имеет вид

$$\mathbf{x}_{n+1} = \mathbf{x}_n + h\mathbf{f}(t_n, \mathbf{x}_n) + \frac{h^2}{2}\left(\frac{\partial \mathbf{f}}{\partial t} + \frac{\partial \mathbf{f}}{\partial \mathbf{x}}\mathbf{f}\right) + \dots$$

Раскладывая (4.2.3) в ряд и приравнивая коэффициенты при соответствующих степенях h, добиваемся того, что формула (4.2.3) задает методы второй степени

$$\mathbf{x}_{n+1} = \mathbf{x}_n + p_1 h\mathbf{f}_n + p_2 h\left(\mathbf{f}_n + \alpha_2 h\frac{\partial \mathbf{f}}{\partial t} + \beta_{21}\frac{\partial \mathbf{f}}{\partial \mathbf{x}}\mathbf{f} + \dots\right),$$
$$p_1 + p_2 = 1, \quad p_2\alpha_2 = 1/2, \quad p_2\beta_{21} = 1/2. \qquad (4.2.4)$$

Условия (4.2.4) представляют собой три уравнения с четырьмя неизвестными и, следовательно, методов второй степени вида (4.2.3) оказывается бесконечно много. В частности, определяя параметры $p_1 = p_2 = 1/2$, $\alpha_2 = \beta_{21} = 1$, получаем метод Эйлера — Коши, а набор $p_1 = 0$, $p_2 = 1$, $\alpha_2 = \beta_{21} = 1/2$ задает усовершенствованный метод ломаных Эйлера (4.2.2). В то же время построить метод третьей степени с двумя вычислениями $\mathbf{f}(t,\mathbf{x})$ не удается.

Увеличивая число вычислений функции $\mathbf{f}(t,\mathbf{x})$ на одном шаге, получаем семейство методов Рунге — Кутты в виде

$$\mathbf{k}_1 = h\mathbf{f}(t_n, \mathbf{x}_n), \quad \mathbf{k}_r = h\mathbf{f}\left(t_n + \alpha_r h, \mathbf{x}_n + \sum_{i=1}^{r-1}\beta_{ri}\mathbf{k}_i\right), \quad r = 1, 2, \dots, s,$$

$$\mathbf{x}_{n+1} = \mathbf{x}_n + \sum_{r=1}^{s} p_r \mathbf{k}_r .$$

Коэффициенты методов вычисляем аналогично тому, как это было выполнено для методов второй степени. При этом если для метода второй степени достаточно рассчитать $\mathbf{f}(t,\mathbf{x})$ два раза на одном шаге, то метод третьей степени требует трех вычислений $\mathbf{f}(t,\mathbf{x})$, а метод четвертой степени — четырех таких вычислений. Все эти методы, как и методы второй степени, образуют семейства. Среди них наиболее популярными являются следующие методы третьей степени:

$$\mathbf{k}_1 = h\mathbf{f}(t_n, \mathbf{x}_n), \quad \mathbf{k}_2 = h\mathbf{f}(t_n + h/2, \mathbf{x}_n + \mathbf{k}_1/2),$$
$$\mathbf{k}_3 = h\mathbf{f}(t_n + h, \mathbf{x}_n - \mathbf{k}_1 + 2\mathbf{k}_2), \quad \mathbf{x}_{n+1} = \mathbf{x}_n + \left(\frac{\mathbf{k}_1 + 4\mathbf{k}_2 + \mathbf{k}_3}{6}\right) \qquad (4.2.5)$$

и четвертой степени:

$$\mathbf{k}_1 = h\mathbf{f}(t_n, \mathbf{x}_n), \quad \mathbf{k}_2 = h\mathbf{f}(t_n + h/2, \mathbf{x}_n + \mathbf{k}_1/2),$$
$$\mathbf{k}_3 = h\mathbf{f}(t_n + h/2, \mathbf{x}_n + \mathbf{k}_2/2), \quad \mathbf{k}_4 = h\mathbf{f}(t_n + h, \mathbf{x}_n + \mathbf{k}_3), \tag{4.2.6}$$
$$\mathbf{x}_{n+1} = \mathbf{x}_n + \left(\frac{\mathbf{k}_1 + 2\mathbf{k}_2 + 2\mathbf{k}_3 + \mathbf{k}_4}{6}\right).$$

Как уже отмечалось, если функция $\mathbf{f}(t,\mathbf{x})$ в (4.0.1) не зависит от $\mathbf{x}$, то все методы интегрирования дифференциальных уравнений превращаются в соответствующие им квадратурные формулы.

Вопрос 16

Каким квадратурным формулам отвечают методы Рунге — Кутты третьей (4.2.5) и четвертой (4.2.6) степени?

С увеличением степени метода резко возрастает число параметров p_r, α_r, β_{ri}, а также число нелинейных уравнений для их определения. Оказывается, что метод Рунге — Кутты четвертой степени является последним методом, у которого количество вычислений $\mathbf{f}(t,\mathbf{x})$ на одном шаге совпадает со степенью метода. Уже метод Рунге — Кутты пятой степени требует вычислять функцию $\mathbf{f}(t,\mathbf{x})$ шесть раз, шестой степени — семь раз, седьмой степени — девять раз, восьмой степени — одиннадцать раз. С дальнейшим ростом степени методов трудности их построения растут по экспоненте. Так в литературе [50] описывается пример метода десятой степени с 17 вычислениями $\mathbf{f}(t,\mathbf{x})$ на шаге!

Теперь обратимся к такому важному моменту, как контроль погрешности метода в процессе интегрирования. Представляется весьма желательным использование переменного шага интегрирования подобно тому, как это делается в программах, реализующих адаптивные квадратурные формулы (например, программа QUANC8 из [14]). Хотелось бы выбирать малый шаг там, где решение меняется быстро, и большой, где оно меняется относительно медленно. Оценивать погрешность по отбрасываемому члену разложения (4.1.10) чрезвычайно неудобно. Поэтому на практике используются различные другие подходы для контроля локальной погрешности методов. Один из них состоит в сравнении на каждом шаге интегрирования решений, получаемых по формулам методов различных степеней. Этот подход реализован в программе RKF45 из [14], построенной на методах Рунге — Кутты — Фель-

берга четвертой и пятой степени. Фельбергу удалось так подобрать параметры методов, что одни и те же шесть вычислений $\mathbf{k}_r$ функции $\mathbf{f}(t,\mathbf{x})$ с различными весами p_r используются для получения решения методами и четвертой, и пятой степени:

$$\mathbf{x}_{n+1}^{(4)} = \mathbf{x}_n + \sum_{r=1}^{6} p_r \mathbf{k}_r, \qquad \mathbf{x}_{n+1}^{(5)} = \mathbf{x}_n + \sum_{r=1}^{6} p_r^* \mathbf{k}_r,$$

$$\mathbf{x}_{n+1}^{(5)} - \mathbf{x}_{n+1}^{(4)} = \sum_{r=1}^{6} \left(p_r^* - p_r\right) \mathbf{k}_r.$$

Тогда разность между этими решениями может использоваться для контроля величины шага дискретности. Программа имеет следующие параметры:

```
RKF45(F, N, X, T, TOUT, RE ,AE, IFLAG, WORK, IWORK)
```

где:

- F — имя процедуры, написанной пользователем для вычисления правых частей системы (4.0.1). Эта программа должна иметь, в свою очередь, следующие параметры:

  ```
  F(T, X, DX)
  ```

  Здесь X — вектор решения в точке T, DX — вектор производных;
- N — количество интегрируемых уравнений;
- X — вектор решения размерностью N в точке T на входе в программу и в точке TOUT при выходе из нее;
- T — начальное значение независимой переменной на входе в программу (при нормальном выходе это TOUT);
- TOUT — точка выхода по независимой переменной;
- RE, AE — границы относительной и абсолютной погрешностей;
- WORK — рабочий вещественный массив размерности 6N + 3;
- IWORK — рабочий целый массив размерности не менее 5;
- IFLAG — указатель режима интегрирования. Обычно при первом обращении на входе IFLAG = 1, а при последующих обращениях на входе IFLAG = 2. Нормальное выходное значение IFLAG = 2. Другие выходные значения указывают на возникшие отклонения от нормального процесса интегрирования:
  - 3 — заданное значение RE оказалось слишком малым и требуется его увеличить;

* 4 — потребовалось более 3000 вычислений $\mathbf{f}(t,\mathbf{x})$ (это отвечает приблизительно 500 шагам). Можно, не изменяя IFLAG, снова обратиться к программе или, если система является жесткой, применить специальные алгоритмы решения жестких систем (о явлении жесткости *см. разд. 4.3*);
* 5 — решение обратилось в нуль, а AE равно нулю. Требуется задать ненулевое значение AE;
* 6 — требуемая точность не достигнута даже при наименьшей допустимой величине шага, и требуется увеличить AE и RE;
* 7 — слишком большое число требуемых выходных точек препятствует выбору естественной величины шага (он может быть значительно увеличен при заданной точности). Нужно или увеличить TOUT-T, или задать значение IFLAG = 2 и продолжить работу программы;
* 8 — неправильное задание параметров процедуры (например, N < 0, AE < 0, RE < 0).

В заключение раздела целесообразно ответить на следующий вопрос.

Вопрос 17

Какой видится перспектива построения программы, аналогичной RKF45, на основе методов Адамса, например, четвертой и пятой степени? Какие дополнительные проблемы здесь придется решать?

4.3. Устойчивость методов. Ограничение на шаг интегрирования и явление жесткости

Подменяя анализ глобальной погрешности анализом локальной, мы высказывали непременное условие такой замены — обеспечение устойчивости решения разностного уравнения метода, которая, в свою очередь, зависит не только от формулы метода, но и от шага интегрирования. В отсутствие возможности оценить глобальную погрешность и устойчивость метода в общем случае появляется желание выбрать некоторую простую модельную ("тестовую") систему уравнений и рассмотреть, как формируется глобальная погрешность для нее.

Такой тестовый пример должен удовлетворять двум требованиям. С одной стороны, он должен быть достаточно простым, чтобы можно было выполнить необходимый анализ, а с другой стороны, он должен быть достаточно "представительным" в том смысле, что сравнительные выводы о свойствах устойчивости различных методов должны носить относительно общий характер и распространяться на значительный круг задач. Этим требованиям удовлетворяет система линейных уравнений

$$\frac{d\mathbf{x}(t)}{dt} = \mathbf{A}\mathbf{x}(t) \tag{4.3.1}$$

с постоянной матрицей. Она является достаточно представительной, т. к. любая нелинейная система может быть линеаризована в окрестности некоторой точки решения и заменена системой (4.3.1). Если какой-либо метод продемонстрирует негативные свойства в смысле устойчивости на примере системы (4.3.1), трудно ожидать от него хороших свойств на нелинейной задаче (обратное, разумеется, не всегда верно).

Пусть все собственные значения λ_k матрицы **A** лежат в левой полуплоскости. Тогда решение системы дифференциальных уравнений (4.3.1) будет асимптотически устойчиво. Чтобы численный метод адекватно отражал реальность, необходимо потребовать, чтобы его разностное уравнение также обладало асимптотически устойчивым решением.

Первоначально обратимся к явному методу ломаных Эйлера. Его формула (4.0.3) применительно к (4.3.1) записывается в виде

$$\mathbf{x}_{n+1} = \mathbf{x}_n + h\mathbf{A}\mathbf{x}_n = \left(\mathbf{E} + h\mathbf{A}\right)\mathbf{x}_n.$$

Для асимптотической устойчивости решения разностного уравнения необходимо, чтобы собственные значения матрицы $\mathbf{E} + h\mathbf{A}$, равные $1 + h\lambda_k$, по модулю были бы меньше единицы $\left(\left|1 + h\lambda_k\right| < 1\right)$, что для вещественных отрицательных λ_k приводит к выполнению неравенства

$$-1 \le 1 + h\lambda_k \le 1, \qquad h\left|\lambda_k\right| < 2. \tag{4.3.2}$$

Для комплексных значений $h\lambda_k = h\alpha_k + jh\omega_k$ условие асимптотической устойчивости $\left(1 + h\alpha_k\right)^2 + h^2\omega_k^2 < 1$ требует, чтобы их значения находились на комплексной плоскости внутри круга с единичным радиусом и центром (–1, 0), а ограничение на шаг имело вид: $h < \dfrac{-2\alpha_k}{\left(\alpha_k^2 + \beta_k^2\right)}$. Множество значений

$h\lambda_k$, удовлетворяющих условию устойчивости разностного уравнения метода, называют *областью устойчивости* данного *метода*. Для явного метода ломаных Эйлера, который является одновременно методом Адамса первой степени и методом Рунге — Кутты первой степени, она представлена на рис. 4.1 (кривая $s = 1$).

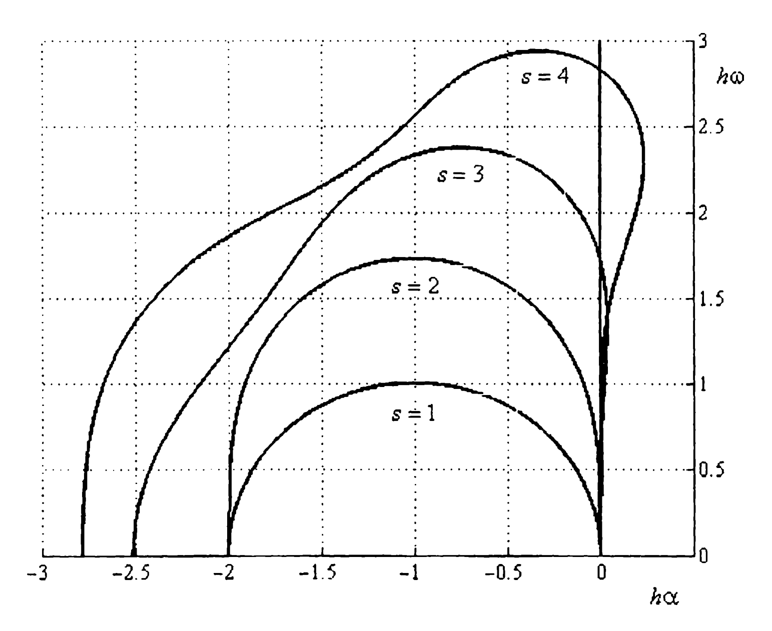


Рис. 4.1. Области устойчивости явных методов Рунге — Кутты (s — степень метода)

Является ли условие (4.3.2) на собственные числа λ_k обременительным для практики? Обратимся к примерам.

Пример 1. Пусть матрица **A** второго порядка имеет два собственных значения: $\lambda_1 = -2$, $\lambda_2 = -1$. Решение (4.3.1) содержит две составляющие с экспонентами $\exp(-2t)$ и $\exp(-t)$. Промежуток наблюдения решения определяется самой медленной экспонентой (три—пять ее постоянных времени): $t \in [0,\ T]$, $T = 3$. Для построения графика такого решения вполне достаточно взять шаг визуализации порядка нескольких десятых и, таким образом, весь график будет содержать примерно десять—двадцать точек. Выполнение условия устойчивости численного метода ломаных Эйлера в соответствии с (4.3.2) приводит к необходимости удовлетворить неравенствам: $h|\lambda_k|_{\max} < 2$

и $h<1$, что вполне согласуется с желаемым шагом построения графика и практически не является обременительным.

Пример 2. Пусть $\lambda_1 = -10^4$, $\lambda_2 = -1$. В решении опять две экспоненты и, т. к. самая медленная осталась такой же, что и в предыдущем примере, время наблюдения решения осталось прежним: $t \in [0,3]$. На этот раз на начальном участке имеется резко меняющаяся экспонента с малой постоянной времени $\tau_1 = \frac{1}{|\lambda_1|} = 10^{-4}$. Для построения графика такого решения необходимо взять десяток точек на начальном участке с шагом $h \sim 10^{-5}$, а затем практически на всем остальном участке решения использовать, как и ранее, шаг порядка нескольких десятых. Однако условие устойчивости (4.3.2) требует, чтобы шаг интегрирования неизменно удовлетворял неравенству и весь промежуток решения содержал несколько десятков тысяч шагов! Любая попытка увеличения шага с нарушением условия (4.3.2) приводит к неустойчивости решения разностного уравнения и незамедлительному "взрыву погрешности". Описанная ситуация будет проявляться тем острее, чем больше разброс между собственными значениями матрицы $\mathbf{A}$.

Системы уравнений (4.3.1) с такими решениями получили название *жестких систем* уравнений. Для них характерны два участка решения. Первый из них малой продолжительностью $\tau_{ПС}$ называют *пограничным слоем*. Решение здесь обладает большими производными и изменяется очень быстро. Второй участок продолжительностью много больше первого $(T \gg \tau_{ПС})$ обладает сравнительно малыми производными, и решение носит относительно спокойный характер.

Всегда ли плохо обусловленная матрица $\mathbf{A}$ системы (4.3.1) порождает жесткую систему? Рассмотрим пример.

Пример 3. Пусть $\lambda_{1,2} = -1 \pm j10^4$, $\lambda_3 = -1$. Число обусловленности весьма велико, но разделение на два участка отсутствует, т. к. на всем отрезке решения наблюдается сильно осциллирующее решение, требующее малого шага визуализации. Для того чтобы плохо обусловленная матрица порождала жесткую систему, необходимо, чтобы большие собственные значения обладали большими по модулю отрицательными вещественными частями, и отвечающие им составляющие решения практически уже не наблюдались за пограничным слоем. Важно и выполнение неравенства $T \gg \tau_{ПС}$. Нелинейные системы (4.0.1) часто относят к числу жестких, когда их матрица Якоби

обладает вышеназванными свойствами на всем протяжении решения. Более строгое определение жестких систем, анализ их свойств и свойств устойчивости численных методов можно найти в книге [40].

Является ли условие (4.3.2) недостатком только явного метода ломаных Эйлера и как ведут себя другие численные методы для жестких систем? К сожалению, все рассмотренные явные методы Рунге — Кутты и Адамса непригодны для решения жестких систем. Так, последовательно применяя методы Рунге — Кутты второй, третьей и четвертой степени (формулы (4.2.1), (4.2.5) и (4.2.6), соответственно) к системе (4.3.1), получаем следующие разностные уравнения:

$$\mathbf{x}_{n+1} = \left(\mathbf{E} + h\mathbf{A} + \frac{h^2\mathbf{A}^2}{2}\right)\mathbf{x}_n,$$

$$\mathbf{x}_{n+1} = \left(\mathbf{E} + h\mathbf{A} + \frac{h^2\mathbf{A}^2}{2} + \frac{h^3\mathbf{A}^3}{6}\right)\mathbf{x}_n,$$

$$\mathbf{x}_{n+1} = \left(\mathbf{E} + h\mathbf{A} + \frac{h^2\mathbf{A}^2}{2} + \frac{h^3\mathbf{A}^3}{6} + \frac{h^4\mathbf{A}^4}{24}\right)\mathbf{x}_n$$

и ограничения на шаг интегрирования, подобно (4.3.2) задающие области устойчивости:

$$\left|1 + h\lambda + \frac{h^2\lambda^2}{2}\right| < 1, \qquad (4.3.3)$$

$$\left|1 + h\lambda + \frac{h^2\lambda^2}{2} + \frac{h^3\lambda^3}{6}\right| < 1, \qquad (4.3.4)$$

$$\left|1 + h\lambda + \frac{h^2\lambda^2}{2} + \frac{h^3\lambda^3}{6} + \frac{h^4\lambda^4}{24}\right| < 1. \qquad (4.3.5)$$

Для вещественных отрицательных λ_k эти ограничения принимают вид:

- $h|\lambda_k| < 2$ — методы второй степени (4.2.1) и (4.2.2);
- $h|\lambda_k| < 2.513$ — метод третьей степени (4.2.5);
- $h|\lambda_k| < 2.785$ — метод Рунге — Кутты четвертой степени (4.2.6).

Для общего случая комплексных значений λ_k области устойчивости представлены на рис. 4.1. Так как все области обладают свойством симметрии

относительно действительной оси, то воспроизводится только часть границы областей, лежащая в верхней полуплоскости. Значения $h\lambda_k$ внутри этих областей удовлетворяют условиям (4.3.3)—(4.3.5). Хотя ограничения на шаг интегрирования незначительно ослабляются при увеличении степени метода, общий объем вычислений при этом даже возрастает, т. к. растет число значений $\mathbf{f}(t,\mathbf{x})$, требуемых на каждом шаге.

Еще хуже обстоят дела с устойчивостью явных методов Адамса. Их разностные схемы

$$\mathbf{x}_{n+1} - \mathbf{x}_n - h\sum_{i=0}^{r-1} b_i \mathbf{f}_{n-i} = 0,$$

примененные к (4.3.1), порождают следующую систему разностных уравнений:

$$\mathbf{x}_{n+1} - \mathbf{x}_n - h\mathbf{A}\sum_{i=0}^{r-1} b_i \mathbf{x}_{n-i} = 0.$$

После замены переменных $\mathbf{x} = \mathbf{U}\mathbf{z}$, где $\mathbf{U}$ — матрица собственных векторов матрицы $\mathbf{A}$, для вектора $\mathbf{z}$ и его компонентов $z^{(k)}$ получаем

$$\mathbf{z}_{n+1} - \mathbf{z}_n - h\mathbf{\Lambda}\sum_{i=0}^{r-1} b_i \mathbf{z}_{n-i} = 0,$$

$$z_{n+1}^{(k)} - z_n^{(k)} - h\lambda_k \sum_{i=0}^{r-1} b_i z_{n-i}^{(k)} = 0, \qquad (4.3.6)$$

$$\mathbf{\Lambda} = \mathbf{U}^{-1}\mathbf{A}\mathbf{U} = \mathrm{diag}(\lambda_1, \lambda_2, \ldots, \lambda_m).$$

Решение разностного уравнения (4.3.6) записывается через корни γ_j его характеристического уравнения

$$\gamma^r - \gamma^{r-1} - h\lambda_k \sum_{i=0}^{r-1} b_i \gamma^{r-1-i} = 0. \qquad (4.3.7)$$

Условие устойчивости (4.3.7) предполагает, что все γ_j должны быть по модулю меньше единицы $|\gamma_j| < 1$. Для вещественных отрицательных значений λ_k необходимым условием этого является выполнение неравенства

$$h|\lambda_k| < \frac{2}{\sum_{i=0}^{r-1} |b_i|},$$

что приводит к еще более серьезным ограничениям на h по сравнению с методами Рунге — Кутты:

- $h|\lambda_k| < 1.0$ — метод Адамса второй степени (4.1.1);
- $h|\lambda_k| < 6/11$ — метод Адамса третьей степени (4.1.2);
- $h|\lambda_k| < 0.3$ — метод Адамса четвертой степени (4.1.3).

В общем случае для построения областей устойчивости методов Адамса может быть использован следующий подход. На границе области устойчивости модуль одного из корней характеристического уравнения (4.3.7) равен единице. Поэтому выполним подстановку $\gamma = \exp(i\varphi)$. Функция

$$F(\varphi) = \frac{\exp(ir\varphi) - \exp(i(r-1)\varphi)}{\sum_{i=0}^{r-1} b_i \exp(i(r-1-i)\varphi)}$$

при изменении φ от 0 до π опишет в плоскости $h\lambda$ кривую, часть которой будет границей области устойчивости рассматриваемых методов. Сами области приведены на рис. 4.2.

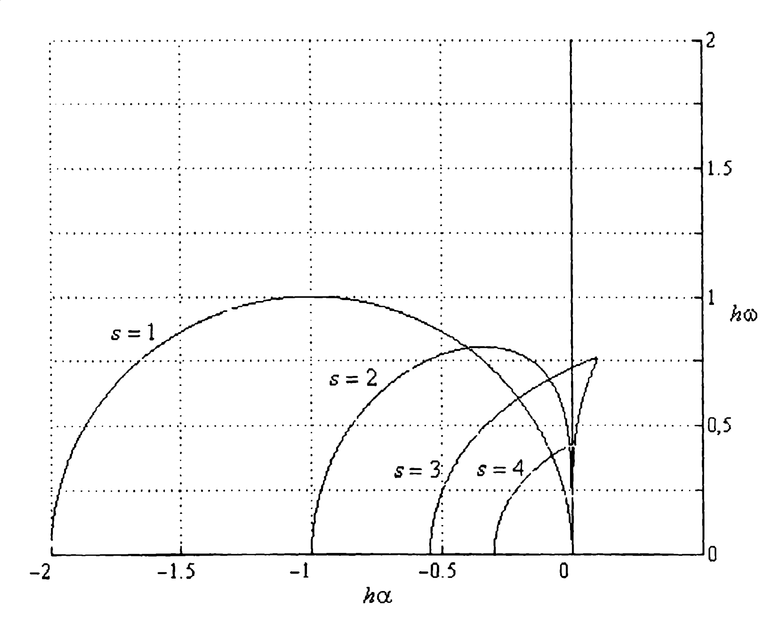


Рис. 4.2. Области устойчивости явных методов Адамса (s — степень точности)

Общая ситуация для вещественных отрицательных значений λ_k складывается следующим образом. Время наблюдения решения T определяется минимальным по модулю собственным значением матрицы **A**, а шаг интегрирования — максимальным. Тогда число шагов N прямо пропорционально числу обусловленности, что и приводит к недопустимым затратам

$$T \sim \frac{1}{|\lambda_k|_{\min}}, \quad h \sim \frac{1}{|\lambda_k|_{\max}}, \quad N = \frac{T}{h} \sim \frac{|\lambda_k|_{\max}}{|\lambda_k|_{\min}} \gg 1.$$

Чтобы изменить ситуацию, для методов, предназначенных интегрировать жесткие системы, следует потребовать, чтобы их область устойчивости включала в себя всю или почти всю левую полуплоскость, что позволит устранить ограничение на шаг типа (4.3.2) и увеличить шаг, когда пограничный слой уже пройден.

Используем для решения (4.3.1) неявный метод ломаных Эйлера (4.0.4). Соответствующее разностное уравнение и ограничение на шаг примут следующий вид:

$$\mathbf{x}_{n+1} = \mathbf{x}_n + h\mathbf{A}\mathbf{x}_{n+1} = (\mathbf{E} - h\mathbf{A})^{-1}\mathbf{x}_n, \quad |1 - h\lambda_k| > 1.$$

Полагая величину λ_k комплексной, $\lambda_k = \alpha_k + j\omega_k$, для области устойчивости метода (рис. 4.3) получим:

$$(1 - h\alpha_k)^2 + h^2\omega_k^2 > 1.$$

Она включает в себя всю левую полуплоскость, и неустойчивость метода проявляется только в круге единичного радиуса с центром (1, 0) (рис. 4.3).

Еще более интересный результат наблюдаем для неявного метода трапеций

$$\mathbf{x}_{n+1} = \mathbf{x}_n + \frac{h}{2}(\mathbf{A}\mathbf{x}_{n+1} + \mathbf{A}\mathbf{x}_n); \quad \left(\mathbf{E} - \frac{h}{2}\mathbf{A}\right)\mathbf{x}_{n+1} = \left(\mathbf{E} + \frac{h}{2}\mathbf{A}\right)\mathbf{x}_n;$$

$$\mathbf{x}_{n+1} = \left(\mathbf{E} - \frac{h}{2}\mathbf{A}\right)^{-1}\left(\mathbf{E} + \frac{h}{2}\mathbf{A}\right)\mathbf{x}_n; \quad \left|\frac{1 + h\lambda_k/2}{1 - h\lambda_k/2}\right| < 1.$$

Тогда для $\lambda_k = \alpha_k + j\omega_k$ получаем

$$\left(1 + \frac{h\alpha}{2}\right)^2 + \frac{h^2\omega^2}{4} < \left(1 - \frac{h\alpha}{2}\right)^2 + \frac{h^2\omega^2}{4},$$

что после упрощений приводит к выполнению неравенства $h\alpha < 0$, т. е. область устойчивости метода совпадает с областью, где устойчивость имеет

место для решения дифференциального уравнения. Таким образом, оба алгоритма могут быть рекомендованы для решения жестких систем. Как будет видно далее, типичное для неявных методов некоторое увеличение трудоемкости одного шага интегрирования с лихвой окупается большим выигрышем в величине шага для жестких систем.

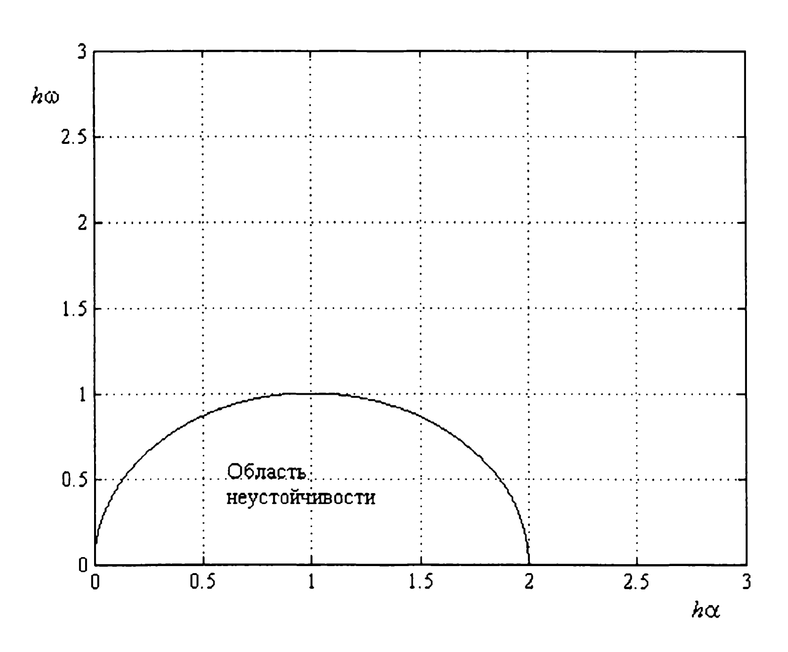


Рис. 4.3. Область устойчивости неявного метода ломаных Эйлера

В заключение рассмотрим, как решается проблема неявного задания $\mathbf{x}_{n+1}$, например, в неявном методе ломаных Эйлера (4.0.4). Решение этого уравнения относительно $\mathbf{x}_{n+1}$ может быть сведено к решению следующей системы:

$$\mathbf{F}(\mathbf{z}) = \mathbf{z} - \mathbf{x}_n - \mathbf{f}(t_{n+1}, \mathbf{z}) = 0 \qquad (4.3.8)$$

методом Ньютона

$$\frac{\partial \mathbf{F}}{\partial \mathbf{z}}\left(\mathbf{z}^{(k)}\right)\left(\mathbf{z}^{(k+1)} - \mathbf{z}^{(k)}\right) = -\mathbf{F}\left(\mathbf{z}^{(k)}\right); \quad \frac{\partial \mathbf{F}}{\partial \mathbf{z}} = \mathbf{E} - h\frac{\partial \mathbf{f}}{\partial \mathbf{z}},$$

где $\mathbf{z}^{(k)}$ — k-е приближение к значению $\mathbf{x}_{n+1}$. Здесь весьма эффективен модифицированный метод Ньютона, когда матрица $\dfrac{\partial \mathbf{F}}{\partial \mathbf{z}}$ вычисляется в точке

$\mathbf{x}_0$, раскладывается в произведение треугольных матриц программой DECOMP, и на последующих итерациях используется только программа SOLVE. Матрица вновь вычисляется только тогда, когда метод Ньютона перестает сходиться за три итерации. Даже если матрица Якоби $\frac{\partial \mathbf{f}}{\partial \mathbf{z}}$ исходной системы уравнений плохо обусловлена, обращение матрицы $\frac{\partial \mathbf{F}}{\partial \mathbf{z}}$, как правило, не вызывает затруднений, т. к. она значительно лучше обусловлена, чем $\frac{\partial \mathbf{f}}{\partial \mathbf{z}}$.

Вопрос 18

Почему последнее утверждение имеет место?

Как уже отмечалось, метод Ньютона, обеспечивая квадратичную скорость сходимости, очень чувствителен к выбору начального приближения. В данном случае с заданием $\mathbf{z}^{(0)}$ проблем не возникает. В качестве $\mathbf{z}^{(0)}$ может быть выбрано значение $\mathbf{x}_n$ или выполнен шаг явным методом ломаных Эйлера

$$\mathbf{z}^{(0)} = \mathbf{x}_n + h\mathbf{f}(t_n, \mathbf{x}_n). \qquad (4.3.9)$$

В последнем случае уместен следующий вопрос.

Вопрос 19

Если исходная система жесткая, а вне пограничного слоя желательно выбрать достаточно большое значение шага h, то возможно ли определять $\mathbf{z}_0$ по (4.3.9) и использовать явный метод ломаных Эйлера, нарушая при этом условие устойчивости (4.3.2)?

В итоге применение метода Ньютона в неявных алгоритмах может быть описано по следующей схеме.

1. В некоторой точке вычисляем матрицу Якоби по аналитическим формулам для ее элементов или с помощью формул численного дифференцирования, а затем производим ее разложение с помощью программы DECOMP.

2. По начальному условию $\mathbf{z}^{(0)}$, рассчитанному с помощью явного метода Эйлера, выполняем итерации метода Ньютона для получения $\mathbf{x}_{n+1}$.

3. После одной-двух итераций по методу Ньютона при достижении сходимости переходим к шагу 2. Возвращение к шагу 1 проводится только в том случае, если метод Ньютона перестает сходиться за три итерации.

Такая организация вычислений приводит к малому объему работы на одном шаге, а сам метод Ньютона с хорошим начальным приближением сходится за одну-две итерации.

По аналогичной схеме для решения жестких систем используются и другие неявные методы. Методы с областью устойчивости, пригодной для решения жестких систем, почти всегда являются неявными, хотя, разумеется, далеко не все неявные методы такую область имеют.

4.4. Численное решение систем линейных дифференциальных уравнений с постоянной матрицей

С системами линейных дифференциальных уравнений

$$\frac{d\mathbf{x}(t)}{dt} = \mathbf{A}\mathbf{x} + \mathbf{b}\,,\ \mathbf{x}(0) = \mathbf{x}_0 \qquad (4.4.1)$$

приходится встречаться крайне часто при построении математических моделей динамических систем. Для их решения может быть применен любой из рассмотренных ранее универсальных алгоритмов, но более эффективным может оказаться тот метод, который учитывает специфику решения (4.4.1). Точное решение системы (4.4.1) имеет вид (П3.14)

$$\mathbf{x}(t) = e^{\mathbf{A}t}\mathbf{x}_0 + \int_0^t e^{\mathbf{A}\tau} d\tau \cdot \mathbf{b}. \qquad (4.4.2)$$

Если решение необходимо представить в виде таблицы при значениях $t_n = nH$, где n — целое число, а H — шаг наблюдения решения, то непосредственное использование формулы (4.4.2) весьма затруднительно. Многократное вычисление матричной экспоненты и интеграла от нее с помощью сходящихся матричных рядов исключительно трудоемко особенно для больших значений t, когда приходится учитывать большое число членов ряда. Использованию формулы Лагранжа — Сильвестра предшествует решение полной проблемы собственных значений и подготовительная работа, связан-

ная с формированием матричных множителей, что также весьма затруднительно при большом размере матрицы $\mathbf{A}$.

Выход из этого положения дает метод, предложенный Ю. В. Ракитским и заключающийся в следующих предварительных матричных преобразованиях.

Запишем решение (4.4.2) в точке $t_{n+1} = t_n + H$ и вычтем из него формулу (4.4.2), предварительно умноженную на $e^{\mathbf{A}H}$:

$$\mathbf{x}(t_n + H) = e^{\mathbf{A}H}\mathbf{x}(t_n) + \left(\int_0^{t_n+H} e^{\mathbf{A}\tau} d\tau - e^{\mathbf{A}H} \int_0^{t} e^{\mathbf{A}\tau} d\tau \right) \mathbf{b} =$$

$$= e^{\mathbf{A}H}\mathbf{x}(t_n) + \left(\int_0^{t_n+H} e^{\mathbf{A}\tau} d\tau - \int_H^{t+H} e^{\mathbf{A}\tau} d\tau \right) \mathbf{b} = \qquad (4.4.3)$$

$$= e^{\mathbf{A}H}\mathbf{x}(t_n) + \int_0^{H} e^{\mathbf{A}\tau} d\tau \cdot \mathbf{b}.$$

В отличие от (4.4.2) запись решения в виде (4.4.3) предполагает лишь *однократное* вычисление матричной экспоненты и интеграла от нее для заданного шага H, а затем последовательное получение $\mathbf{x}(t_n)$ пошаговым методом. Для вычисления матричной экспоненты и интеграла от нее воспользуемся матричным степенным разложением

$$e^{\mathbf{A}H} \cong \mathbf{E} + H\mathbf{A} + \frac{H^2\mathbf{A}^2}{2} + \ldots, \quad \int_0^H e^{\mathbf{A}\tau} d\tau \cong H\left(\mathbf{E} + \frac{H\mathbf{A}}{2} + \ldots \right). \qquad (4.4.4)$$

Чтобы разобраться со скоростью сходимости этого ряда, обратимся к скалярному ряду

$$e^{-0.1} \cong 1 - 0.1 + 0.01/2 - 0.001/6 + \ldots;$$

$$e^{-10} \cong 1 - 10 + 100/2 - 1000/6 + \ldots$$

Ясно, что при больших по модулю показателях экспоненты ряд сходится крайне медленно. Если матрица $\mathbf{A}$ имеет простую структуру, то она сама и ее матричная экспонента могут быть приведены к диагональной форме

$$e^{\mathbf{A}H} = \mathbf{U} \begin{pmatrix} e^{\lambda_1 H} & \ldots & 0 \\ \ldots & \ldots & \ldots \\ 0 & \ldots & e^{\lambda_m H} \end{pmatrix} \mathbf{U}^{-1},$$

где на диагонали стоят скалярные ряды и скорость сходимости всего матричного ряда определяется скоростью сходимости скалярного с максимальным по модулю показателем $|\lambda_k|_{\max} H$. Для удовлетворительной скорости сходимости необходимо обеспечить выполнение неравенства $|\lambda_k|_{\max} H < 1$ или достаточного условия для нормы матрицы $\|\mathbf{A}\| H < 1$.

Если система (4.4.1) является жесткой, то вне пограничного слоя появляется потребность выбрать шаг наблюдения решения H таким, чтобы $|\lambda_k|_{\max} H \gg 1$, а это приходит в противоречие с условием эффективного построения матричного ряда. Поэтому по заданному желаемому значению H выбирается такое целое число N, что для $h = \frac{H}{2^N}$ имеет место неравенство $\|\mathbf{A}\| h < 1$. Далее строим $e^{\mathbf{A}h}$ разложением в матричный ряд, а затем N раз возводим матричную экспоненту в квадрат по формуле $e^{2\mathbf{A}h} = e^{\mathbf{A}h} \cdot e^{\mathbf{A}h}$, достигая матрицы $e^{\mathbf{A}H}$.

Аналогичная формула удвоения шага может быть получена и для вектора

$$\mathbf{g}(h) = \int_0^h e^{\mathbf{A}\tau} d\tau \cdot \mathbf{b}, \quad \mathbf{g}(2h) = \left(\mathbf{E} + e^{\mathbf{A}h}\right)\mathbf{g}(h).$$

Эта формула удвоения шага легко устанавливается после преобразований

$$\mathbf{g}(2h) = \int_0^{2h} e^{\mathbf{A}\tau} d\tau \cdot \mathbf{b} = \int_0^h e^{\mathbf{A}\tau} d\tau \cdot \mathbf{b} + \int_h^{2h} e^{\mathbf{A}\tau} d\tau \cdot \mathbf{b} =$$
$$= \int_0^h e^{\mathbf{A}\tau} d\tau \cdot \mathbf{b} + e^{\mathbf{A}h} \int_0^h e^{\mathbf{A}\tau} d\tau \cdot \mathbf{b} = \left(\mathbf{E} + e^{\mathbf{A}h}\right)\mathbf{g}(h).$$

Таким образом, алгоритм решения системы (4.4.1) записывается такой последовательностью действий:

1. Задаемся желаемым значением H. Выбираем целое число N, такое, что

$$h = \frac{H}{2^N} < \frac{1}{\|\mathbf{A}\|}.$$

2. Для шага h строим матричную экспоненту $e^{\mathbf{A}h}$ и вектор $\mathbf{g}(h)$ разложением в ряд с небольшим числом членов.

3. Получаем матрицу $e^{\mathbf{A}H}$ и вектор $\mathbf{g}(H)$, используя N раз формулу удвоения шага.
4. Решаем уравнение (4.4.3) пошаговым методом.

На основе изложенного алгоритма написана программа LSODE [41] со следующими параметрами:

```
LSODE (N, H, CH, A, B, X, EAH, SL, INDEX)
```

где:

- N — размерность системы;
- H — шаг наблюдения решения;
- CH — константа для оценки начального шага h (рекомендуемое значение для обычных систем — 0,1, а для жестких — порядка 5.0);
- A, B — матрица и вектор системы (4.4.1);
- X — вектор решения;
- EAH — матрица, содержащая элементы матрицы $e^{\mathbf{A}H}$;
- SL — рабочий массив размерности N;
- INDEX — управляющий параметр с входными значениями:
  - –1 (первое обращение к программе, вектор B нулевой, т. е. решается однородная система);
  - –2 (первое обращение к программе, вектор B может быть ненулевым);
  - 0 (не первое обращение к программе).

Нормальное выходное значение — 0.

4.5. Решение краевой задачи. Методы стрельбы и конечных разностей

В рассмотренной ранее задаче Коши одно из решений системы (4.0.1) выделяется заданием начальных условий (t_0, $\mathbf{x}_0$). Однако это не единственный способ. Задавая условия при двух или более значениях независимой переменной, приходим к краевой задаче.

В общем случае краевые (граничные) условия выглядят следующим образом:

$$\Phi_i\left(x^{(1)}(t_k), ..., x^{(m)}(t_k)\right)=0, \quad a \le t_k \le b, \quad 1 \le i \le m.$$

В зависимости от вида уравнения и краевых условий можно провести классификацию краевых задач, схожую с классификацией задач Коши. Важным подклассом являются линейные краевые задачи, когда и система (4.0.1) и краевые условия являются линейными. Эти условия имеют вид:

$$\alpha_i x^{(1)}(t_k)+\beta_i x^{(2)}(t_k)+...+\omega_i(t_k)x^{(m)}(t_k)=a_i, \quad i \le m.$$

Линейная краевая задача является однородной, если однородны уравнения и краевые условия. Такая задача всегда имеет тривиальное решение $\mathbf{x}(t)\equiv 0$, и в этом случае представляет интерес отыскание нетривиальных решений.

В свою очередь, из краевых задач выделяют двухточечные, когда условия задаются на левом и правом концах промежутка, т. е. при $t_k = a$ и $t_k = b$. Например, дифференциальное уравнение второго порядка

$$-\frac{d}{dt}\left(p(t)\frac{dx(t)}{dt}\right)+q(t)x(t)=f(t),$$

$$t\in[0,\ 1], \quad p(t)\ge p_0>0, \quad q(t)>0,$$

где краевые условия $x(0)=x(1)=0$ определяют задачу, которая является моделью многих физических процессов: распределение тепла в неоднородном стержне, распределение концентрации вещества в процессах диффузии и др.

Несмотря на разнообразие форм краевых условий, краевые задачи в основном решаются одними и теми же численными методами. Выделяют два основных подхода:

- сведение к многократному решению задачи Коши;
- сведение к решению алгебраических систем.

Второй подход включает в себя как конечно-разностные, так и проекционные методы. К последним относятся, в свою очередь, давно применяющиеся методы коллокаций, Галеркина, Ритца, а также метод конечных элементов.

Многократное решение задачи Коши демонстрирует метод стрельбы, имеющий аналогию со стрельбой, когда, зафиксировав недолет или перелет, угол стрельбы изменяют так, чтобы следующий выстрел был ближе к цели.

Рассмотрим систему из двух уравнений

$$\frac{du}{dt} = f_1(t,u,v), \qquad \frac{dv(t)}{dt} = f_2(t,u,v), \qquad t \in [a,\ b]$$

с граничными условиями

$$\Phi_1\left(u(a), v(a)\right) = 0, \qquad \Phi_2\left(u(b), v(b)\right) = 0.$$

Выберем произвольное значение $u_a = u(a)$ и выразим из первого краевого условия $v(a) = \xi(u_a)$. С начальными условиями u_a и v_a проинтегрируем систему каким-либо методом. Результатом будут функции $u(t, u_a)$ и $v(t, u_a)$, зависящие от u_a как от параметра. При подстановке $u(t, u_a)$ и $v(t, u_a)$ в правое краевое условие получаем функцию относительно u_a. Задача свелась к нахождению решения уравнения $L(u_a) = 0$, где $L(u_a) = \Phi_2\left(u(b, u_a), v(b, u_a)\right)$.

Конкретный вид функции $L(u_a)$ неизвестен, но значения ее для любых значений u_a легко вычисляются, и это дает возможность воспользоваться любым методом нахождения корней нелинейного уравнения. Так, в частности, можно воспользоваться уже известной процедурой-функцией `ZEROIN(A, B, F, EPS)`. Последовательность вызова отдельных процедур отражает рис. 4.4.

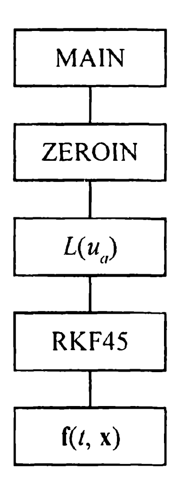


Рис. 4.4. Последовательность вызова процедур в методе стрельбы

Основная программа вызывает `ZEROIN`. Предварительно подбираются два значения $u_a = A$ и $u_a = B$, для которых функция $L(u_a)$ имеет различный

знак. Программа ZEROIN вызывает функцию $L(u_a)$, которую программирует пользователь. Эта функция, в свою очередь, обращается к программе RKF45, которая вызывает процедуру, вычисляющую $\mathbf{f}(t,\mathbf{x})$. Значение функции $L(u_a)$ для заданного u_a определяется выражением

$$L(u_a) = \Phi_2\left(u(b,u_a),v(b,u_a)\right).$$

Описанная ситуация резко упрощается, если задача является линейной двухточечной краевой

$$\frac{du}{dt} = p_1(t)u(t) + q_1(t)v(t) + s_1(t);$$

$$\frac{dv}{dt} = p_2(t)u(t) + q_2(t)v(t) + s_2(t);$$

с граничными условиями

$$\alpha_1 u(a) + \beta_1 v(a) = r_1;$$

$$\alpha_2 u(b) + \beta_2 v(b) = r_2.$$

Тогда из первого условия имеем $v(a) = \left(r_1 - \alpha_1 u(a)\right)/\beta_1$. В силу линейности задачи решение будет зависеть от u_a линейно, и функция $L(u_a)$ также будет линейной. Отсюда следует, что для вычисления u_a^* — левого начального условия для функции $u(t)$, дающего решение краевой задачи, достаточно дважды проинтегрировать систему до $t = b$ с двумя различными начальными условиями (u_a^1, v_a^1) и (u_a^2, v_a^2), найти $L(u_a^1)$ и $L(u_a^2)$ и линейной интерполяцией определить u_a^*. Таким образом, u_a^* будет корнем уравнения

$$\frac{u_a - u_a^2}{u_a^1 - u_a^2} L\left(u_a^1\right) + \frac{u_a - u_a^1}{u_a^2 - u_a^1} L\left(u_a^2\right) = 0.$$

Другой подход к решению краевых задач заключается в их сведении к решению систем линейных и нелинейных уравнений. Здесь одними из наиболее популярных являются конечно-разностные методы, иллюстрацию работы которых проведем на примере линейной краевой задачи:

$$\frac{d^2 y}{dt^2} + p(t)\frac{dy}{dt} + q(t)y(t) = f(t); \qquad t \in [a,\ b],$$

$$\alpha_1 y(a) + \beta_1 \frac{dy(a)}{dt} = \gamma_1, \qquad \alpha_2 y(b) + \beta_2 \frac{dy(b)}{dt} = \gamma_2.$$

Вводя дискретные значения независимой переменной

$$t_k = t_0 + k \cdot h, \quad h = \frac{b-a}{N}, \quad t_0 = a, \quad t_N = b, \quad y_k = y(t_k),$$

запишем в этих точках исходное уравнение, предварительно аппроксимировав первую и вторую производные по формулам численного дифференцирования:

$$\frac{dy(t_k)}{dt} \approx \frac{y_{k+1} - y_{k-1}}{2h}; \qquad \frac{d^2y(t_k)}{dt^2} \approx \frac{y_{k+1} - 2y_k + y_{k-1}}{h^2}.$$

В результате получим систему линейных алгебраических уравнений относительно y_k

$$a_k y_{k-1} + b_k y_k + c_k y_{k+1} = -h^2 f_k, \quad k = 1, 2, \ldots, N-1, \tag{4.5.1}$$

где

$$a_k = \frac{hp_k}{2} - 1; \qquad b_k = 2 - h^2 q_k; \qquad c_k = -\frac{hp_k}{2} - 1.$$

Граничные условия в такой ситуации принимают вид:

$$\begin{aligned} &\alpha_1 y_0 + \beta_1 \frac{-y_2 + 4y_1 - 3y_0}{2h} = \gamma_1; \\ &\alpha_2 y_N + \beta_2 \frac{3y_N - 4y_{N-1} + y_{N-2}}{2h} = \gamma_2. \end{aligned} \tag{4.5.2}$$

Здесь использовались формулы численного дифференцирования (1.12.3) и (1.12.5) для дифференцирования в начале и в конце таблицы. Выразим с помощью (4.5.2) значения y_0 и y_N и подставим их в первое и последнее уравнения системы (4.5.1). После приведения подобных слагаемых каждое из этих двух уравнений будет содержать только два неизвестных

$$\tilde{b}_1 y_1 + \tilde{c}_1 y_2 = -h^2 \tilde{f}_1; \qquad \tilde{a}_{N-1} y_{N-2} + \tilde{b}_{N-1} y_{N-1} = -h^2 \tilde{f}_{N-1}. \tag{4.5.3}$$

Уравнения (4.5.1) для $k = 2, 3, \ldots, N-2$ совместно с (4.5.3) порождают линейную систему с трехдиагональной матрицей

$$\begin{pmatrix} \tilde{b}_1 & \tilde{c}_1 & 0 & 0 & \ldots & 0 \\ a_2 & b_2 & c_2 & 0 & \ldots & 0 \\ 0 & a_3 & b_3 & c_3 & \ldots & 0 \\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\ 0 & \ldots & 0 & a_{N-2} & b_{N-2} & c_{N-2} \\ 0 & \ldots & 0 & 0 & \tilde{a}_{N-1} & \tilde{c}_{N-1} \end{pmatrix}.$$

При этом можно использовать естественный порядок исключения неизвестных. Модификация метода Гаусса применительно к таким системам получила название *метода прогонки*. Получим расчетные формулы метода.

Решение системы (4.5.1) и (4.5.3) будем искать в следующем виде:

$$y_k = \Theta_k y_{k+1} + \Psi_k; \qquad k = 1, 2, ..., N-2. \tag{4.5.4}$$

Отсюда выразим y_{k-1} через y_{k+1}:

$$y_{k-1} = \Theta_{k-1} y_k + \Psi_{k-1} = \Theta_{k-1}\Theta_k y_{k+1} + \left(\Theta_{k-1}\Psi_k + \Psi_{k-1}\right) \tag{4.5.5}$$

и подставим выражения (4.5.4) и (4.5.5) в (4.5.1):

$$\left[\Theta_k\left(a_k\Theta_{k-1} + b_k\right) + c_k\right] y_{k+1} + \left[\Psi_k\left(a_k\Theta_{k-1} + b_k\right) + a_k\Psi_{k-1} + h^2 f_k\right] = 0.$$

Приравнивая отдельно каждую квадратную скобку нулю, получаем уравнения для получения Θ_k и Ψ_k:

$$\begin{aligned} &\Theta_k = -\frac{c_k}{b_k + a_k\Theta_{k-1}}, \qquad &&\Psi_k = -\frac{a_k\Psi_{k-1} + h^2 f_k}{b_k + a_k\Theta_{k-1}}, \\ &\Theta_1 = -\frac{\tilde{c}_1}{\tilde{b}_1}, &&\Psi_1 = -\frac{h^2\tilde{f}_1}{\tilde{b}_1}. \end{aligned} \tag{4.5.6}$$

Выражения для Θ_1 и Ψ_1 получаются непосредственным сравнением (4.5.4) с первым уравнением (4.5.3). Нахождение коэффициентов Θ_k и Ψ_k по формулам (4.5.6) называется *прямой прогонкой*. Теперь, когда эти коэффициенты найдены, по формуле (4.5.4) последовательно определяем y_k, начиная с y_{N-1}. Для задания y_{N-1} запишем (4.5.4) для $k = N-2$ и воспользуемся вторым уравнением (4.5.3):

$$y_{N-2} = \Theta_{N-2} y_{N-1} + \Psi_{N-2};$$

$$\tilde{a}_{N-1} y_{N-2} + \tilde{b}_{N-1} y_{N-1} = -h^2 \tilde{f}_{N-1}.$$

Решая эти два уравнения совместно, для y_{N-1} получаем

$$y_{N-1} = -\frac{\tilde{a}_{N-1}\Psi_{N-2} + h^2\tilde{f}_{N-1}}{\tilde{b}_{N-1} + \tilde{a}_{N-1}\Theta_{N-2}}. \tag{4.5.7}$$

Нахождение y_k по формулам (4.5.4) и (4.5.7) называется *обратной прогонкой*.

Можно показать, что достаточным условием успешного применения метода прогонки является выполнение следующих условий для коэффициентов системы (4.5.1):

$$a_k \neq 0,\ \ c_k \neq 0,\quad |b_k| \geq |a_k| + |c_k|,\quad k = 1, 2, \ldots, N-2;\quad |\Theta_1| \leq 1,\quad |\tilde{a}_{N-1}| < |\tilde{b}_{N-1}|.$$

Число арифметических операций в традиционном методе Гаусса для большой заполненной матрицы пропорционально примерно кубу ее размера, в то время как рассмотренный метод обладает трудоемкостью, зависящей от размера лишь линейно. Поэтому для решения краевых задач целесообразно иметь специальную программу метода прогонки, как альтернативу DECOMP и SOLVE .

В заключение отметим, что нелинейные краевые задачи аналогичными приемами сводятся к решению некоторой уже нелинейной системы уравнений.

4.6. Решение краевой задачи. Введение в проекционные методы

В данном разделе кратко ознакомимся с идеями методов коллокации, Галеркина и конечных элементов. В основе их построения лежит аппроксимация решения дифференциального уравнения конечной линейной комбинацией базисных функций. При этом ограничимся иллюстрацией на примере линейной двухточечной краевой задачи

$$\frac{d}{dt}\left(p(t)\frac{d}{dt}y(t)\right) + q(t)y(t) = f(t);\quad t \in [0,\ 1]$$

с простейшими краевыми условиями $y(0) = y(1) = 0$.

Приближенное решение будем искать в виде

$$y(t) = \sum_{k=1}^{n} c_k \varphi_k(t),$$

где базисные функции $\varphi_1(t),\ \varphi_2(t),\ \ldots,\ \varphi_n(t)$ удовлетворяют граничным условиям задачи: $\varphi_k(0) = \varphi_k(1) = 0$.

Первоначально обратимся к *методу* коллокации. На отрезке $[0,\ 1]$ выберем n точек $t_1,\ t_2,\ \ldots,\ t_n$. Потребуем, чтобы приближенное решение в этих точках удовлетворяло уравнениям

$$\frac{d}{dt}\left(p(t)\frac{d}{dt}\sum_{k=1}^{n} c_k \varphi_k(t)\right)\Bigg|_{t = t_m} + q(t_m)\sum_{k=1}^{n} c_k \varphi_k(t_m) = f(t_m),\quad m = 1, \ldots, n.$$

После дифференцирования и группировки слагаемых получим линейную систему алгебраических уравнений для нахождения коэффициентов c_k, являющихся компонентами вектора $\mathbf{c} = (c_1, c_2, ..., c_n)^T$:

$$\mathbf{Ac} = \mathbf{f}, \qquad \sum_{k=1}^{n} a_{mk} c_k = f(t_m), \qquad m = 1, ..., n;$$

$$a_{mk} = p(t_m)\varphi_k''(t_m) + p'(t_m)\varphi_k'(t_m) + q(t_m)\varphi_k(t_m).$$

Вопрос 20

Из какого предыдущего раздела заимствована основная идея метода коллокации?

Перейдем к *методу Галеркина*. Здесь коэффициенты c_k приближенного решения выбираются таким образом, чтобы невязка приближенного решения $r(t)$

$$r(t) = \frac{d}{dt}\left(p(t)\frac{d}{dt}y(t)\right) + q(t)y(t) - f(t)$$

была бы ортогональна всем базисным функциям

$$(r(t), \varphi_m(t)) = 0.$$

Переписав скалярное произведение в соответствии с его определением в пространстве непрерывных функций

$$\int_0^1 \left(\frac{d}{dt}\left(p(t)\frac{d}{dt}y(t)\right) + q(t)y(t) - f(t)\right)\varphi_m(t)dt = 0,$$

подставив в интеграл искомый вид решения и поменяв местами суммирование и дифференцирование, получаем также систему линейных алгебраических уравнений относительно c_k:

$$\mathbf{Ac} = \mathbf{f}, \qquad \sum_{k=1}^{n} a_{mk} c_k = \int_0^1 f(t)\varphi_m(t), \qquad m = 1, ..., n;$$

$$a_{mk} = \int_0^1 \left(\frac{d}{dt}\left(p(t)\frac{d}{dt}\varphi_k(t)\right) + q(t)\varphi_k(t)\right)\varphi_m(t)dt.$$

Вопрос 21

А с каким предыдущим разделом перекликается основная идея метода Галеркина?

Интегрируя по частям первое слагаемое под знаком интеграла для a_{mk}, имеем:

$$a_{mk} = -\int_0^1 p(t) \frac{d\varphi_k(t)}{dt} \frac{d\varphi_m(t)}{dt} dt + \int_0^1 q(t)\varphi_k(t)\varphi_m(t)dt.$$

Такое представление коэффициентов a_{mk} уже не требует, чтобы приближенное решение (и базисные функции) обладали вторыми производными. Это обстоятельство является также весьма важным в одном из вариантов метода конечных элементов, о котором будет сказано позже.

Примерами базисных функций, удовлетворяющих краевым условиям, могут служить:

- $\varphi_k(t) = \sin(k\pi)$;
- $\varphi_k(t) = t^k(1-t)$; $\varphi_k(0) = \varphi_k(1) = 0$; $k = 1, 2, ..., n$.

По поводу систем уравнений, возникающих в методах коллокации и Галеркина, следует сделать два замечания. Во-первых, матрицы коэффициентов в обоих методах в общем случае являются заполненными. Напомним, что, например, в методе конечных разностей матрица была трехдиагональной.

Во-вторых, преимущество метода Галеркина перед методом коллокации в задаче из данного раздела в том, что его матрица симметрична.

В заключение обратимся к методу конечных элементов. Это проекционный метод со специфическими базисными или координатными функциями. Ими являются *B-сплайны* (базисные сплайны), обладающие таким свойством, что любой сплайн-полином на отрезке отыскания решения может быть представлен линейной комбинацией B-сплайнов.

В качестве базисных функций в методе Галеркина воспользуемся кусочно-линейными функциями вида

$$\varphi_k(t) = \begin{cases} (t - t_{k-1})/h, & t_{k-1} \le t \le t_k; \\ (t_{k+1} - t)/h, & t_k \le t \le t_{k+1}; \\ 0, t > t_{k+1}, & t < t_{k-1}. \end{cases}$$

Это линейные B-сплайны, представляющие собой так называемые "функции-крышки". Их производные — кусочно-постоянные функции:

$$\frac{d}{dt}\varphi_k(t)=\begin{cases}1/h, & t_{k-1}\le t\le t_k;\\ -1/n, & t_k\le t_{k+1};\\ 0, & t>t_{k+1},\ t<t_{k-1}.\end{cases}$$

Наличие разрыва в точках t_{k-1}, t_k, t_{k+1} не влияет на интегрирование, которое проводится отдельно по каждому участку разбиения:

$$a_{mk}=\sum_{i=1}^{n-1}\int_{t_i}^{t_{i+1}}\left(-p(t)\frac{d\varphi_k(t)}{dt}\frac{d\varphi_m(t)}{dt}+q(t)\varphi_k(t)\varphi_m(t)\right)dt.$$

Произведения $\frac{d\varphi_k(t)}{dt}\frac{d\varphi_m}{dt}$ и $\varphi_k(t)\varphi_m(t)$ по определению функций $\varphi_k(t)$ тождественно равны нулю всегда, кроме $k-1\le m\le k+1$, т. е. $a_{mk}=0$, если $|k-m|>1$. Это означает, что не равны нулю лишь коэффициенты a_{kk}, $a_{k,k-1}$, $a_{k,k+1}$, т. е. матрица **A** линейной системы трехдиагональная, как и при применении конечно-разностного метода.

Естественно, с одной стороны, возможно использование сплайнов более высоких степеней, например кубических, что должно привести к уменьшению ошибки аппроксимации. С другой стороны, вычисление коэффициентов a_{mk} по методу коллокации проще, поскольку не требует интегрирования, как в методе Галеркина, и поэтому метод коллокации с B-сплайнами является рабочим методом, где матрица линейной системы также трехдиагональная. Выбор метода для конкретной краевой задачи — сравнительно простого конечно-разностного или более трудоемкого проекционного — требует анализа специфики самой задачи.

4.7. Введение в методы решения уравнений в частных производных

Наличие в реальных процессах не одной, а нескольких независимых переменных приводит к математическим моделям, описываемым дифференциальными уравнениями в частных производных.

Наиболее часто встречающиеся типы этих уравнений второго и первого порядка проиллюстрируем простыми примерами:

- *уравнение переноса*

$$\frac{\partial u}{\partial t} = -a\frac{\partial u}{\partial x} + f(t,x);$$

- *уравнение теплопроводности* (диффузии), называемое уравнением *параболического* типа

$$\frac{\partial u}{\partial t} = a\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + \frac{\partial^2 u}{\partial z^2}\right);$$

- *волновое уравнение*, называемое уравнением гиперболического типа

$$\frac{\partial^2 u}{\partial t^2} = a\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + \frac{\partial^2 u}{\partial z^2}\right);$$

- *уравнение потенциала* (уравнения Пуассона, Лапласа), называемое уравнением *эллиптического* типа

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + \frac{\partial^2 u}{\partial z^2} = f(x,y,z).$$

Эти названия отражают суть типичных физических явлений и процессов, описываемых приведенными уравнениями.

Как и в случае обыкновенных дифференциальных уравнений, для выделения одного решения необходимо задать начальные и (или) краевые (граничные) условия. Здесь выделяют *условия Дирихле*, когда задается начальное распределение функции $u(t, x, y, z)$ на границах области ее изменения, *условия Неймана*, когда определяется изменение функции $u(t, x, y, z)$ на границе, а также их различные комбинации.

Проблемы, возникающие при решении уравнений в частных производных, настолько обширны, а постановки задач настолько разнообразны, что в данном разделе остановимся лишь на некоторых простейших подходах к их решению. При этом чаще всего исходная проблема сводится к уже известным задачам решения обыкновенных дифференциальных или алгебраических уравнений.

В качества первого примера рассмотрим одномерный случай уравнения теплопроводности с соответствующими начальными и граничными условиями

$$\frac{\partial u}{\partial t} = a\frac{\partial^2 u}{\partial x^2}, \qquad u(0,x) = f(x), \qquad u(t,0) = u(t,1) = 0.$$

Введя дискретизацию по переменной x с обозначениями $x_k = k\Delta x$; $\Delta x = 1/n$; $x_0 = 0$; $x_n = 1$ и используя для второй производной простейшую формулу численного дифференцирования, получим систему уже обыкновенных дифференциальных уравнений относительно $u_k(t) = u(t, x_k)$:

$$\frac{du_k(t)}{dt} = \frac{a}{(\Delta x)^2}\left(u_{k+1}(t) - 2u_k(t) + u_{k-1}(t)\right); \quad k = 1, 2, \ldots, n-1.$$

При этом $u_0(t) = 0$, $u_n(t) = 0$, а начальные условия определяются равенствами $u_k(0) = f(x_k)$, $k = 1, \ldots, n-1$.

Матрица полученной линейной системы имеет специфический трехдиагональный вид, а ее собственные значения, определяющие характер поведения отдельных составляющих решения, в этом случае могут быть определены аналитически и рассчитываются по формулам:

$$\lambda_k = -\frac{2a}{(\Delta x)^2}\left(1 + \cos\frac{k\pi}{n}\right); \quad k = 1, \ldots, n-1.$$

Для числа обусловленности матрицы имеет место выражение

$$k(\mathbf{A}) = \frac{|\lambda_k|_{\max}}{|\lambda_k|_{\min}} = \frac{\lambda_1}{\lambda_{n-1}} = \frac{1 + \cos(\pi/n)}{1 + \cos(\pi(n-1)/n)} = \frac{1 + \cos(\pi/n)}{1 - \cos(\pi/n)}.$$

При больших значениях n это отношение примерно равно $4n^2/\pi^2$.

Таким образом, уменьшение величины Δx, с одной стороны, повышает точность аппроксимации, а с другой — делает систему более жесткой со всеми вытекающими из этого проблемами. Такой способ сведения уравнения в частных производных к системе обыкновенных дифференциальных уравнений получил название *метода прямых*.

Второй вариант этого метода основан на идеях проекционного подхода. Для той же задачи с теми же граничными условиями будем искать решение в виде

$$u^*(x,t) = \sum_{k=0}^{n} c_k(t)\varphi_k(x),$$

где $\varphi_0(x)$, $\varphi_1(x)$, ..., $\varphi_n(x)$ — базисные функции, удовлетворяющие граничным условиям, т. е. $\varphi_k(0) = \varphi_k(1) = 0$. Как и ранее, вводя дискретизацию

по x, потребуем, чтобы в точках x_i функция $u^*(x,t)$ в таком виде удовлетворяла точно дифференциальному уравнению

$$\sum_{k=0}^{n} \frac{dc_k(t)}{dt} \varphi_k(x_i) = \sum_{k=0}^{n} ac_k(t) \frac{d^2\varphi_k(x_i)}{dx^2}, \quad i = 0, 1, \ldots, n.$$

В векторной форме эта система приобретает вид:

$$\mathbf{\Phi} \frac{d\mathbf{c}(t)}{dt} = a\mathbf{\Phi}''\mathbf{c}(t); \quad \frac{d\mathbf{c}(t)}{dt} = a\mathbf{\Phi}^{-1}\mathbf{\Phi}''\mathbf{c}(t); \quad \mathbf{c}(t) = (c_0, \ldots, c_n)^{\mathrm{T}}.$$

$$\mathbf{\Phi} = \begin{pmatrix} \varphi_0(x_0) & \ldots & \varphi_n(x_0) \\ \ldots & \ldots & \ldots \\ \varphi_0(x_n) & \ldots & \varphi_n(x_n) \end{pmatrix}, \quad \mathbf{\Phi}'' = \begin{pmatrix} \varphi_0''(x_0) & \ldots & \varphi_n''(x_0) \\ \ldots & \ldots & \ldots \\ \varphi_0''(x_n) & \ldots & \varphi_n''(x_n) \end{pmatrix}.$$

Начальные условия $c_k(0)$ получают из равенства $u^*(x_i, 0) = f(x_i)$.

Таким образом, согласно методу прямых исходное уравнение в частных производных сводится к системе обыкновенных дифференциальных уравнений с начальными условиями. При этом используется программное обеспечение, разработанное для решения задачи Коши.

Еще один подход к решению уравнений в частных производных предполагает дискретизацию по всем независимым переменным. В качестве примера рассмотрим одномерное волновое уравнение (гиперболического типа)

$$\frac{\partial^2 u}{\partial r^2} = a\frac{\partial^2 u}{\partial x^2}, \quad 0 \le x \le 1; \quad t \ge 0$$

с начальными и граничными условиями

$$u(x,0) = f(x), \quad \frac{\partial u(x,0)}{\partial t} = g(x), \quad u(0,t) = \alpha, \quad u(1,t) = \beta.$$

Технология метода предполагает замену производных в уравнении и начальном условии двусторонними и односторонними формулами численного дифференцирования. Прямоугольную сетку по x и t выберем для простоты равномерной: $t_n = n\Delta t$, $x_m = m\Delta x$, $u_{m,n} = u(x_m, t_n)$,

$$\left.\frac{\partial^2 u}{\partial t^2}\right|_{x_m, t_n} \approx \frac{u_{m,n+1} - 2u_{m,n} + u_{m,n-1}}{(\Delta t)^2}; \quad n = 1, 2, \ldots, N-1;$$

$$\left.\frac{\partial^2 u}{\partial x^2}\right|_{x_m, t_n} \approx \frac{u_{m+1,n} - 2u_{m,n} + u_{m-1,n}}{(\Delta x)^2}; \quad n = 1, 2, \ldots, N-1.$$

6 Зак. 3418

Решение находится последовательно по временным слоям для фиксированных значений t_n на заданной сетке. Нулевой слой использует начальное условие $u_{m,0} = f(x_m)$. На первом слое учитывается второе начальное условие:

$$\frac{u_{m,1} - u_{m,0}}{\Delta t} \approx g(x_m); \quad n = 1, 2, ..., N-1.$$

Начиная со второго временного слоя, используются основные рабочие формулы, учитывающие аппроксимацию вторых производных по x и t:

$$u_{m,n+1} = 2u_{m,n} - u_{m,n-1} + a\left(\frac{\Delta t}{\Delta x}\right)^2 \left(u_{m+1,n} - 2u_{m,n} + u_{m-1,n}\right).$$

Таким образом, решение будет представлено таблицами функции $u(x_m, t_n)$ для каждого t_n на временной сетке с шагом Δt.

Используемые аппроксимации производных свидетельствуют о втором порядке точности разностной схемы как по времени, так и по пространственной переменной. Не вдаваясь в подробности, приведем условие устойчивости разностной схемы, связывающее значение двух шагов дискретизации

$$\Delta t \le \frac{\Delta x}{\sqrt{a}}.$$

Проиллюстрированный на простом примере подход применим, разумеется, и к более общим задачам для одного или нескольких уравнений. При этом необходимо постоянно помнить о возможном возникновении неустойчивости, т. к. провести соответствующий анализ часто не представляется возможным.

Ранее рассматривались лишь простейшие аппроксимации производных и явные методы. Значительно более эффективными являются неявные методы. В частности, неявный метод получим, если при записи системы уравнений вторая производная по пространственной координате будет отнесена не к точке t_n, как в явном методе, а к точке t_{n+1}. Аналогом может служить неявный метод ломаных Эйлера при решении обыкновенных дифференциальных уравнений.

Теперь для того, чтобы найти решение послойно, необходимо, в отличие от явного метода, для каждого слоя решать систему уравнений относительно $u_{m,n+1}$:

$$u_{m,n+1} = 2u_{m,n} - u_{m,n-1} + a\left(\frac{\Delta t}{\Delta x}\right)^2 \left(u_{m+1,n+1} - 2u_{m,n+1} + u_{m-1,n+1}\right).$$

Вводя обозначение $\mu = a\left(\frac{\Delta t}{\Delta x}\right)^2$ и перенося все известные слагаемые в правую часть, получаем

$$-\mu u_{m-1,n+1} + (1+2\mu)u_{m,n+1} - \mu u_{m+1,n+1} = 2u_{m,n} - u_{m.n-1}.$$

При записи первого и последнего уравнения необходимо учесть граничные условия: $u_{0,n+1} = \alpha$; $u_{M,n+1} = \beta$, которые в данном случае те же, что и в явном методе. Таким образом, для каждого слоя решаем систему с одной и той же трехдиагональной матрицей (что учитывается). Дополнительные потери во времени для одного слоя по сравнению с явным методом чаще всего оборачиваются существенным выигрышем в устойчивости, что позволяет иметь временную сетку гораздо более редкой и значительно сократить общие вычислительные затраты.

Наконец, рассмотрим конечно-разностный метод для уравнения Пуассона (эллиптический тип):

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = f(x,y).$$

В качестве области исключительно для простоты будем рассматривать единичный квадрат $0 \le x \le 1$, $0 \le y \le 1$ на плоскости (x,y) с равномерной сеткой и одинаковым шагом h по обеим пространственным координатам. Здесь $h = \frac{1}{N+1}$, где N — количество внутренних узлов сетки. В качестве граничных условий на сторонах квадрата рассмотрим условия Дирихле: $u(x,y) = g(x,y)$, где g — заданная функция двух переменных.

Используя по-прежнему формулы численного дифференцирования для вторых производных, сведем исходное уравнение к системе линейных алгебраических уравнений для всех внутренних точек области:

$$-u_{i,j-1} - u_{i-1,j} + 4u_{i,j} - u_{i+1,j} - u_{i,j+1} = -f_{i,j}; \qquad u_{i,j} = u(ih, jh).$$

При этом на границе выполняются условия:

$$u_{0,j} = g(0,y_j), \quad u_{N+1,j} = g(1,y_j), \quad u_{i,0} = g(x_i,0), \quad u_{i,N+1} = g(x_i,1).$$

Соотношения для внутренних узлов образуют систему из N^2 уравнений, которую можно записать в матричной форме $\mathbf{Az} = \mathbf{b} - h^2\mathbf{f}$, где

$$\mathbf{A} = \begin{pmatrix} \mathbf{B}_N & -\mathbf{E}_N & \dots & 0 & 0 \\ -\mathbf{E}_N & \mathbf{B}_N & \dots & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & \mathbf{B}_N & -\mathbf{E}_N \\ 0 & 0 & \dots & -\mathbf{E}_N & \mathbf{B}_N \end{pmatrix}, \quad \mathbf{B} = \begin{pmatrix} 4 & -1 & .. & 0 & 0 \\ -1 & 4 & \dots & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & 4 & -1 \\ 0 & 0 & \dots & -1 & 4 \end{pmatrix},$$

а $\mathbf{E}_N$ — единичная матрица порядка $N \times N$.

Компоненты векторов $\mathbf{z}$, $\mathbf{b}$, $\mathbf{f}$ определены следующим образом:

$$\mathbf{z} = \left(u_{1,1}, \dots, u_{N,1}, u_{1,2}, \dots, u_{N,2}, \dots, u_{1,N}, \dots, u_{N,N}\right)^T,$$

$$\mathbf{f} = \left(f_{1,1}, \dots, f_{N,1}, f_{1,2}, \dots, f_{N,2}, \dots, f_{1,N}, \dots, f_{N,N}\right)^T,$$

$$\mathbf{b} = \left(\mathbf{b}_1, \dots, \mathbf{b}_N\right)^{\mathrm{T}}, \quad \mathbf{b}_1 = \left(u_{0,1} + u_{1,0}, u_{2,0}, \dots, u_{N-1,0}, u_{N,0} + u_{N+1,1}\right),$$

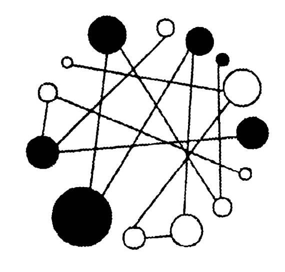
$$\mathbf{b}_i = \left(u_{0,i}, 0, 0, \dots, u_{N+1,i}\right), \quad i = 2, \dots, N-1;$$

$$\mathbf{b}_N = \left(u_{0,N} + u_{1,N+1}, u_{2,N+1}, \dots, u_{N-1,N+1}, u_{N,N+1} + u_{N+1,N}\right).$$

Полученная матрица является разреженной. При $N = 100$ имеем 10^4 уравнений и 10^8 элементов $\mathbf{A}$. Вместе с тем, каждая строка $\mathbf{A}$ содержит не более пяти отличных от нуля элементов. Для подобных матриц разрабатываются специальные способы их хранения и обработки в памяти компьютера, позволяющие эффективно решать задачи такого класса.

Для решения уравнений в частных производных (особенно эллиптического и параболического типа), устойчивое распространение имеют проекционные методы и, в частности, метод конечных элементов. В этой книге данный подход был проиллюстрирован на решении краевых задач для обыкновенных дифференциальных уравнений. Здесь отметим только, что он успешно используется и для задач данного раздела.

ГЛАВА 5



Введение в минимизацию функций

В практических приложениях часто возникает задача определения максимума или минимума вещественной функции $f\left(x^{(1)}, \ldots, x^{(n)}\right)$ от n вещественных переменных на множестве $\mathbf{S}$ n-мерного пространства (такие функции называют *функционалами*). В дальнейшем ограничимся лишь поиском минимума, т. к. при возникновении задачи на максимум достаточно сменить знак у минимизируемой функции. Если множество $\mathbf{S}$ совпадает со всем n-мерным пространством, то задачу минимизации называют *безусловной*. В противном случае задача имеет ограничения (обычно в виде совокупности нелинейных функций $g^{(i)}$, удовлетворяющих равенствам или неравенствам)

$$g^{(i)}\left(x^{(1)}, \ldots, x^{(n)}\right) \geq 0, \quad i = 1, 2, \ldots, p,$$

определяющие множество $\mathbf{S}$.

Задачи с ограничениями, как правило, много сложнее задач безусловной минимизации. Если функция f и ограничения $g^{(i)}$ являются линейными функциями, то говорят о задаче линейного программирования. Если f или какая-либо $g^{(i)}$ нелинейная, имеем задачу нелинейного программирования. В данной главе ограничимся знакомством с некоторыми подходами к решению задач безусловной минимизации. Более того, как и в разделе, посвященном решению нелинейных уравнений, рассмотрим методы нахождения локальных минимумов, попутно отметив, что в настоящее время нахождение глобального минимума уже для $n > 2$ часто представляет непреодолимые трудности.

5.1. Минимизация функции одной переменной

Предположим, что для $f(x)$, определенной на $[a,\ b]$, имеется единственное значение, отвечающее минимуму f на $[a,\ b]$. При этом для $x > x^*$ функция $f(x)$ строго возрастает и для $x < x^*$ строго убывает. Такая функция называется *унимодальной*, ее минимум может достигаться как внутри промежутка $[a, b]$, так и на его краях. В дальнейшем речь будет идти именно об унимодальных на заданном промежутке функциях.

Обозначим за ε требуемую точность определения x^*. Тогда задача минимизации состоит в сокращении промежутка неопределенности $[a,\ b]$ до $[\tilde{a},\ \tilde{b}]$ так, чтобы $x^* \in [\tilde{a}, \tilde{b}]$ и $\tilde{b} - \tilde{a} < \varepsilon$.

Если функция вычислена в некоторых точках x_0 и x_1 промежутка неопределенности, то возможны три варианта расположения минимума и сокращения промежутка неопределенности:

- $f(x_0) > f(x_1)$, отбрасываем $[a, x_0]$, $x_1 < x^*$ (рис. 5.1, *а*);
- $f(x_0) < f(x_1)$, отбрасываем $[x_1, b]$, $x_0 > x^*$ (рис. 5.1, *б*);
- $f(x_0) = f(x_1)$, отбрасываем $[a, x_0]$ и $[x_1, b]$, $x_0 \le x^* \le x_1$ (рис. 5.1, *в*).

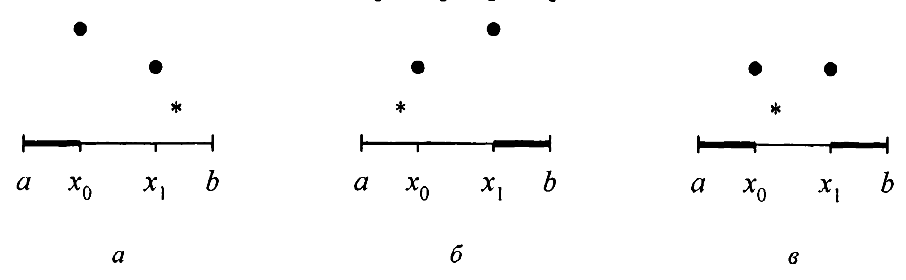


Рис. 5.1. Варианты расположения минимума

Отсюда очевидно, что значение $f(x)$ необходимо вычислить, как минимум, в двух точках внутри промежутка $[a,\ b]$, чтобы последний мог быть сокращен.

Простейший *метод половинного деления* на очередном шаге вычисляет функцию в двух точках: $x_0 = \frac{a+b}{2} - \frac{\varepsilon}{2}$ и $x_1 = \frac{a+b}{2} + \frac{\varepsilon}{2}$, с точностью до $\frac{\varepsilon}{2}$ совпадающих с серединой $[a, b]$. Если, например, $f(x_0) < f(x_1)$, то промежуток сокращается почти в два раза, и процедура повторяется уже на $[a, x_1]$. Следует отметить, что значение $f(x_0)$ уже вычислено, и если бы точка x_0 находилась в оптимальном для последующего поиска положении, то очередное сокращение интервала неопределенности могло бы достигаться ценой лишь одного дополнительного вычисления $f(x)$. Метод половинного деления не использует эту возможность, и точка x_0 отбрасывается, т. к. расположена слишком близко к правому концу промежутка $[a, x_1]$. Повысить эффективность алгоритма позволяет следующая постановка задачи.

Разрешено вычислить $f(x)$ заданное число раз n. Необходимо использовать эти вычисления так, чтобы максимально сократить длину промежутка возможного нахождения x^*. При этом достаточно очевидно, что на каждом шаге две точки внутри промежутка следует располагать симметрично относительно его середины, т. к. нет оснований предпочитать одну из половин промежутка.

Пусть на некотором шаге имеем промежуток длиной F_k. Расстояние от левого конца промежутка до внутренней правой точки x_{k-1} обозначим за F_{k-1}, а до левой x_{k-2} — за F_{k-2}. Если $f(x_{k-2}) < f(x_{k-1})$, то длина нового интервала сокращается до F_{k-1}, точка x_{k-2} становится уже правой, а новая левая точка x_{k-3} должна быть уже расположена так, чтобы x_{k-3} и x_{k-2} были симметрично расположены относительно середины промежутка. Это условие симметричности порождает очевидное соотношение для старого и нового интервалов:

$$F_k = F_{k-1} + F_{k-2}, \quad F_{k-1} = F_{k-2} + F_{k-3}. \tag{5.1.1}$$

На последнем шаге наиболее выгодна ситуация, когда приходим к некоторому интервалу неопределенности длиной F_2, причем точка x_1, унаследованная с предыдущего шага, расположена ровно в середине ($F_1 = F_2/2$). Тогда, вычислив $f(x)$ в точке x_0, почти совпадающей с x_1 $\left(F_0 \approx F_1, |x_0 - x_1| < \varepsilon\right)$, сократим промежуток практически в два раза.

Числа F_k, называемые *числами Фибоначчи* и вычисляемые в соответствии с (5.1.1), определяют наиболее эффективное разбиение промежутка и оптимальное решение задачи. Значения F_0 и F_1 задаются единичными:

$$F_0 = F_1 = 1, \quad F_2 = 2, \quad F_3 = 3, \quad F_4 = 5, \quad F_5 = 8, \ldots$$

На первом шаге длина исходного промежутка $[a, b]$ уменьшается в $\frac{F_n}{F_{n-1}}$ раз, а внутренние точки имеют координаты:

$$x_{n-1} = a + (b-a)F_{n-1}/F_n \quad \text{и} \quad x_{n-2} = a + (b-a)F_{n-2}/F_n.$$

На последующих шагах длина интервала неопределенности сокращается в

$$\frac{F_{n-1}}{F_{n-2}}, \ \frac{F_{n-2}}{F_{n-3}}, \ \ldots, \ \frac{F_2}{F_1}$$

раз соответственно. Напомним, что разбиение промежутка с использованием чисел Фибоначчи оптимально тогда, когда заранее известно количество n вычислений f. Длина исходного интервала при этом сокращается в F_k раз с точностью до ε.

Если число n заранее неизвестно, то числа Фибоначчи перестают определять оптимальное решение задачи. Условие симметричности (5.1.1) должно сохранять свой вид, однако требование $F_0 = F_1 = 1$ должно быть заменено. Вместо него потребуем, чтобы две внутренние точки делили промежуток в одной и той же пропорции на каждом шаге минимизации для каждого нового интервала. Такое условие "повторяемости ситуации" приобретает вид

$$\alpha = \frac{F_{n-1}}{F_n} = \frac{F_{n-2}}{F_{n-1}} = \frac{(F_n - F_{n-1})}{F_{n-1}} = \frac{1}{\alpha} - 1$$

или $\alpha^2 + \alpha - 1 = 0$ и $\alpha^* = (\sqrt{5} - 1)/2 \approx 0{,}618$. Этот метод получил название *метод золотого сечения* (рис. 5.2) и для своего очередного шага не требует знания F_n, поскольку для любого промежутка $[a, b]$ внутренние точки всегда имеют координаты: $x_{\text{лев}} = b - \alpha^*(b-a)$ и $x_{\text{прав}} = a + \alpha^*(b-a)$. Зависимость алгоритма от n очевидна, т. к. на каждом шаге промежуток сокращается ровно в $1/\alpha^*$ раз.

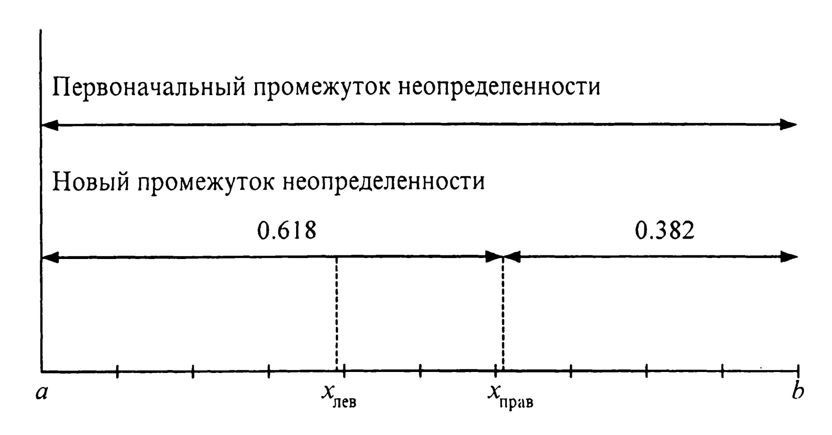


Рис. 5.2. Метод золотого сечения

Если количество n вычислений функции $f(x)$ известно заранее, то метод минимизации с числами Фибоначчи сокращает исходный промежуток в F_n раз, а метод золотого сечения в $(1/\alpha^*)^{n-1}$ раз (на первом шаге этого метода функция вычисляется дважды). Можно заметить, что при больших значениях частное этих двух величин стабилизируется. Для объяснения последнего факта запишем точное решение разностного уравнения (5.1.1) с единичными начальными условиями, определяющими числа Фибоначчи:

$$F_n = \frac{1}{\sqrt{5}}\left(\frac{1+\sqrt{5}}{2}\right)^{n+1} - \frac{1}{\sqrt{5}}\left(\frac{1-\sqrt{5}}{2}\right)^{n+1} = \frac{1}{\sqrt{5}}\left(\frac{1}{\alpha^*}\right)^{n+1} - \frac{1}{\sqrt{5}}\left(-\alpha^*\right)^{n+1}$$

С ростом n второе слагаемое решения стремится к нулю, что дает возможность записать $\lim\limits_{n\to\infty} \frac{F_n}{F_{n-1}} = \frac{1}{\alpha^*}$.

Метод золотого сечения аналогично методу бисекции поиска корней нелинейного уравнения на каждом шаге сокращает промежуток в $1/\alpha^*$ раз *независимо* от вида унимодальной функции.

Вопрос 22

Такое гарантированное сокращение промежутка независимо от вида функции является достоинством или недостатком алгоритма?

Извлечь возможные выгоды из предполагаемой гладкости функции позволяет *метод последовательной параболической интерполяции*. Этот алгоритм на каждом шаге ориентируется на три точки: x_{k-2}, x_{k-1}, x_k. По ним строится парабола (интерполяционный полином второй степени) и вершина этой параболы (точка минимума) определяет значение x_{k+1}, которое заменяет x_{k-2}.

Сочетание методов последовательной параболической интерполяции и золотого сечения демонстрирует процедура-функция `FMIN (A, B, F, EPS)`, идеология которой весьма созвучна с программой `ZEROIN` для поиска нуля функции. Используя второй метод, программа по возможности переключается на первый. Как и в `ZEROIN`, параметры `A` и `B` определяют интервал поиска, `F` — вычисляемую функцию, а `EPS` — границу погрешности. Есть, однако, важное различие между этими программами, которое влияет на выбор величины ε, задающей параметр `EPS` в `FMIN`. Если $f(x)=0$, а $f'(x)\neq 0$, то для малых ε имеем

$$f(x+\varepsilon)=f(x)+\varepsilon f'(x)+\varepsilon^2 f''(x)/2+\ldots\approx \varepsilon f'(x).$$

Малые изменения в x вызывают пропорционально малые изменения в $f(x)$, поэтому разумно выбирать границу погрешности для `ZEROIN` примерно той же величины, что и ошибки в значениях функции (зачастую это величины порядка ошибки округления в данном компьютере).

Если же мы ищем точку минимума, где $f'(x)=0$, а $f''(x)\neq 0$, то для малых ε получаем:

$$f(x+\varepsilon)=f(x)+\varepsilon f'(x)+\varepsilon^2 f''(x)/2+\ldots=f(x)+\varepsilon^2 f''(x)/2.$$

Изменение порядка ε в x теперь вызывает изменение в $f(x)$ порядка ε^2, поэтому разумно выбирать границу погрешности для `FMIN` не меньше, чем квадратный корень из ошибки в значениях функции. Иными словами, если простые нули функции можно находить почти с полной машинной точностью, то точки минимума — лишь с половинной.

5.2. Введение в многомерную минимизацию

Теперь обратимся к постановке задачи безусловной минимизации функции многих переменных $f(\mathbf{x})=f\left(x^{(1)}, \ldots, x^{(m)}\right)$. Если эта функция непрерывна

вместе со своими частными производными, то задача минимизации может быть сведена к решению системы нелинейных уравнений

$$\operatorname{grad} f(\mathbf{x}) = \left(\frac{\partial f}{\partial x^{(1)}}, \frac{\partial f}{\partial x^{(2)}}, \dots, \frac{\partial f}{\partial x^{(m)}} \right)^T = 0, \qquad (5.2.1)$$

задающих нулевое значение компонентам вектора градиента $f(\mathbf{x})$. С другой стороны, пусть задана некоторая система нелинейных уравнений

$$\varphi_k(\mathbf{x}) = \varphi\left(x^{(1)}, x^{(2)}, \dots, x^{(m)}\right) = 0, \qquad k = 1,\ 2,\ \dots,\ m. \qquad (5.2.2)$$

Вместо ее решения можно минимизировать некоторую функцию

$$f(\mathbf{x}) = \sum_{k=1}^{m} \varphi_k^2(\mathbf{x}). \qquad (5.2.3)$$

Ее минимум равен нулю и достигается при тех же значениях $\mathbf{x}$, которые удовлетворяют системе (5.2.2). (Строго говоря, здесь могут появиться посторонние локальные экстремумы.) Минимизация (5.2.3) часто проводится специальными методами, учитывающими особый вид минимизируемой функции.

Поскольку между методами минимизации функций и методами решения нелинейных уравнений существует тесная связь, типовой является следующая ситуация. Если имеется хорошее начальное приближение к точке минимума минимизируемой функции $f(\mathbf{x})$, то целесообразно перейти к решению системы (5.2.1), например, методом Ньютона с квадратичной скоростью сходимости. С другой стороны, при решении нелинейных уравнений (5.2.2) в отсутствие хорошего начального приближения целесообразно получить его, используя первоначально методы минимизации применительно к функции (5.2.3).

Для дальнейшего рассмотрения различных методов минимизации введем некоторые обозначения и определения. Пусть $\mathbf{x}^*$ — точка минимума $f(\mathbf{x})$, а $\mathbf{x}_n$ — n-е приближение к $\mathbf{x}^*$. Множество значений вектора $\mathbf{x}$, удовлетворяющих уравнению

$$f(\mathbf{x}) = c_n,$$

где c_n — некоторая константа, будем называть *поверхностью уровня (линией уровня)* функции $f(\mathbf{x})$.

Зададимся некоторым вектором $\mathbf{v}_n$. Он может выбираться сравнительно произвольно, но не должен быть направлен по касательной к поверхности уровня в точке $\mathbf{x}_n$. Тогда новое приближение может быть определено по формуле

$$\mathbf{x}_{n+1} = \mathbf{x}_n + h\mathbf{v}_n. \tag{5.2.4}$$

Обязательно найдется такое значение h (положительное или отрицательное), при котором $f(\mathbf{x}_{n+1}) < f(\mathbf{x}_n)$. Различные методы отличаются друг от друга выбором вектора $\mathbf{v}_n$ и величины h. В частности, при заданном векторе $\mathbf{v}_n$ значение h может выбираться из условия минимума функции

$$\Psi(h) = f\left(\mathbf{x}_n + h\mathbf{v}_n\right).$$

Задача свелась, таким образом, к минимизации функции одной переменной $\Psi(h)$, и здесь могут быть использованы любые методы, в том числе из предыдущего раздела. Вместе с тем, следует отметить, что отсутствует практическая необходимость в нахождении точного минимума функции $\Psi(h)$, требующего многократного вычисления $f(\mathbf{x})$. Целесообразнее удовлетвориться некоторым значением h таким, что $f(\mathbf{x}_{n+1}) < f(\mathbf{x}_n)$, сменить вектор $\mathbf{v}_n$ на $\mathbf{v}_{n+1}$ и продолжить минимизацию в другом направлении. Рассмотрим различные варианты выбора вектора $\mathbf{v}_n$.

Если в качестве $\mathbf{v}_n$ выбирать компоненты вектора $\mathbf{x}$, то имеем *методы покоординатного спуска*. Если назначать все компоненты $x^{(k)}$ в строго определенном порядке (циклически), получим *обычный (циклический) покоординатный спуск*. Когда же выбор очередного компонента носит случайный характер, имеем *случайный покоординатный спуск*. Наконец, если на каждом шаге в качестве $\mathbf{v}_n$ выбирается тот компонент, в направлении которого $f(\mathbf{x})$ убывает быстрее всего, получаем *релаксационный покоординатный спуск*.

В последнем случае нужная координата будет определяться номером того компонента вектора градиента, который максимален по модулю. Вместе с тем, если вектор градиента необходимо строить, то целесообразно использовать уже *методы* не покоординатного, а *градиентного спуска* (скорейшего спуска), когда в качестве $\mathbf{v}_n$ выбирается вектор антиградиента и формула метода для положительного значения h приобретает вид

$$\mathbf{x}_{n+1} = \mathbf{x}_n - h\,\mathbf{grad}\,f(\mathbf{x}_n). \tag{5.2.5}$$

При таком подходе к минимизации использование оптимального значения h

$$f(\mathbf{x}_{n+1}) = \min_{h>0} f(\mathbf{x}_n - h\,\mathbf{grad}\, f(\mathbf{x}_n))$$

приводит к *методу наискорейшего градиентного спуска*.

Название последнего метода создает иллюзию, что метод с лучшими свойствами создан быть не может. Действительно, вектор антиградиента $\left(-\mathbf{grad}\, f(\mathbf{x}_n)\right)$ является направлением наибольшего убывания функции, *но... в окрестности данной точки* $\mathbf{x}_n$. Это направление в m-мерном пространстве отнюдь не обязательно указывает на точку минимума функции $f(\mathbf{x}_n)$ и слишком часто бывает весьма далеким от желаемого. Примером, когда вектор градиента не является удачным направлением для минимизации даже для функции лишь от двух переменных, может служить так называемая *тестовая функция Розенброка*

$$f(x^{(1)}, x^{(2)}) = 100\left(x^{(2)} - x^{(1)}x^{(1)}\right)^2 + \left(1 - x^{(1)}\right)^2 \qquad (5.2.6)$$

с начальным значением $\mathbf{x}_0 = (-0.5,\ 0.5)^T$ и точкой минимума $\mathbf{x}^* = (1,\ 1)^T$. Нормированный вектор антиградиента в точке $\mathbf{x}_0$ имеет вид:

$$\mathbf{v}_0 = -\mathbf{grad}\, f(\mathbf{x}_0) = (-0.68491,\ -0.72863)^T.$$

Движение в этом направлении явно не способствует приближению к $\mathbf{x}^*$, что легко видеть по значению $\mathbf{x}_1$, полученному методом наискорейшего градиентного спуска:

$$\mathbf{x}_1 \approx (-0.612,\ 0.381)^T.$$

Справедливости ради следует заметить, что и методы покоординатного спуска демонстрируют для этой функции крайне медленную сходимость к точке минимума.

Если имеется хорошее начальное приближение к $\mathbf{x}^*$, то для минимизации функции $f(\mathbf{x})$ может быть использован метод Ньютона (3.2.3), примененный к системе нелинейных уравнений (5.2.1), отражающих необходимое условие экстремума.

Матрица Якоби применительно к вектору градиента приобретает вид:

$$\mathbf{H}=\frac{\partial^2 f}{\partial \mathbf{x}^2}=\begin{pmatrix} \frac{\partial^2 f}{\partial x^{(1)}\partial x^{(1)}} & \frac{\partial^2 f}{\partial x^{(1)}\partial x^{(2)}} & \cdots & \frac{\partial^2 f}{\partial x^{(1)}\partial x^{(m)}} \\ \frac{\partial^2 f}{\partial x^{(2)}\partial x^{(1)}} & \frac{\partial^2 f}{\partial x^{(2)}\partial x^{(2)}} & \cdots & \frac{\partial^2 f}{\partial x^{(2)}\partial x^{(m)}} \\ \cdots & \cdots & \cdots & \cdots \\ \frac{\partial^2 f}{\partial x^{(m)}\partial x^{(1)}} & \frac{\partial^2 f}{\partial x^{(m)}\partial x^{(2)}} & \cdots & \frac{\partial^2 f}{\partial x^{(m)}\partial x^{(m)}} \end{pmatrix} \quad (5.2.7)$$

и называется матрицей Гессе минимизируемой функции $f(\mathbf{x})$. Сам метод Ньютона минимизации функций в этих обозначениях выглядит следующим образом:

$$\mathbf{H}(\mathbf{x}_{n+1}-\mathbf{x}_n)=-\mathbf{grad}\, f(\mathbf{x}_n). \quad (5.2.8)$$

Естественно, что он обладает достоинствами и недостатками, отмеченными нами в *разд. 3.1* для алгоритмов (3.1.5) и (3.2.3). Среди них одним из важнейших является сочетание высокой скорости сходимости с требованием хорошего начального приближения. Возможен вариант и модифицированного метода Ньютона с постоянной в течение нескольких итераций матрицей $\mathbf{H}$.

Однако есть и одно важное отличие матрицы Гессе в формуле (5.2.7) по сравнению с произвольной матрицей Якоби в (3.2.3), заключающееся в симметричности $\mathbf{H}$. Напомним ряд характерных свойств симметрических матриц.

1. Симметрические матрицы имеют только вещественные собственные значения (см. теорему 2 в *разд. П3.3*).
2. Собственные векторы симметрической матрицы, отвечающие различным собственным значениям, ортогональны (см. теорему 3 в *разд. П3.3*).
3. Если матрица симметрическая и $(\mathbf{x},\ \mathbf{y})$ — скалярное произведение, то $(\mathbf{Ax},\mathbf{y})=(\mathbf{x},\mathbf{Ay})$ *(подробнее см. разд. П3.2)*.
4. Из теоремы 3 в *разд. П3.5* следовало, что матрица простой структуры (количество линейно независимых собственных векторов равно размеру матрицы) подобна диагональной матрице $\mathbf{\Lambda}$: $\mathbf{A}=\mathbf{U\Lambda U}^{-1}$, причем на диагонали $\mathbf{\Lambda}$ стоят собственные значения матрицы $\mathbf{A}$. Для симметрической матрицы формула диагонализации приобретает более простой вид. Действительно, пусть столбцами матрицы $\mathbf{U}$ являются собственные векторы $\mathbf{u}_k$

матрицы $\mathbf{A}$, нормированные так, что $(\mathbf{u}_k, \mathbf{u}_k) = 1$. Тогда $\mathbf{U}^T\mathbf{U} = \mathbf{E}$ и $\mathbf{U}^T = \mathbf{U}^{-1}$. Матрицу $\mathbf{U}$ с таким свойством называют *ортогональной*. Формула диагонализации в такой ситуации принимает вид: $\mathbf{A} = \mathbf{U}\Lambda\mathbf{U}^T$.

5.3. Явление овражности и дифференциальное уравнение линии спуска

При интегрировании нелинейных дифференциальных уравнений возникала ситуация, когда крайне сложно было анализировать глобальную погрешность в общем случае и свойства различных методов изучались на примере "тестовой" системы линейных уравнений с постоянной матрицей. Такая "тестовая" система должна была быть, с одной стороны, достаточно простой, а с другой — сравнительно "представительной" в том смысле, что сделанные выводы о свойствах различных методов должны были носить широкий характер и распространяться на значительный круг задач.

Нечто подобное возникает и при сравнении различных методов минимизации. Какова же здесь "тестовая" функция? Гладкая функция одной переменной в окрестности точки минимума может быть неплохо описана параболой, а функция многих переменных — квадратичной функцией:

$$f(\mathbf{x}) = \frac{1}{2}(\mathbf{A}\mathbf{x}, \mathbf{x}) + (\mathbf{b}, \mathbf{x}) + c = \frac{1}{2}\sum_{i=1}^{m}\sum_{k=1}^{m} a_{ik}x^{(i)}x^{(k)} + \sum_{i=1}^{m} b^{(i)}x^{(i)} + c, \qquad (5.3.1)$$

связанной в m-мерном пространстве с поверхностями второго порядка. Здесь $\mathbf{A}$ — заданная симметрическая матрица; $\mathbf{b}$ — постоянный вектор; c — константа. Выражения для градиента, матрицы Гессе и точки минимума $\mathbf{x}^*$ минимизируемой функции (6.3.1) имеют вид:

$$\mathbf{grad}\, f(\mathbf{x}) = \mathbf{A}\mathbf{x} + \mathbf{b}, \qquad \frac{\partial^2 f}{\partial \mathbf{x}^2} = \mathbf{A}, \qquad \mathbf{x}^* = -\mathbf{A}^{-1}\mathbf{b}, \qquad (5.3.2)$$

где $\mathbf{x}^*$ — точка минимума $f(\mathbf{x})$.

Теперь выясним, как выглядят линии уровня квадратичной функции (5.3.1), удовлетворяющие уравнению $f(\mathbf{x}) = \text{const}$. Первоначально выполним замену

переменных в (5.3.1) $\mathbf{x} = \mathbf{y} - \mathbf{A}^{-1}\mathbf{b}$, отвечающую лишь переносу начала координат:

$$f = \frac{1}{2}(\mathbf{Ay}, \mathbf{y}) + c^*, \quad c^* = c - \frac{1}{2}\left(\mathbf{b}, \mathbf{A}^{-1}\mathbf{b}\right).$$

Следующая замена переменных $\mathbf{y} = \mathbf{Uz}$ с ортогональной матрицей $\mathbf{U}$, столбцами которой являются нормированные собственные векторы $\mathbf{A}$, означает лишь поворот осей без растяжения:

$$f = \frac{1}{2}(\mathbf{AUz}, \mathbf{Uz}) + c^* = \frac{1}{2}\left(\mathbf{U}^T\mathbf{AUz}, \mathbf{z}\right) + c^* =$$

$$= \frac{1}{2}(\mathbf{\Lambda z}, \mathbf{z}) + c^* = \frac{1}{2}\sum_{k=1}^{m}\left(\frac{z^{(k)}}{d_k}\right)^2 + c^*, \qquad (5.3.3)$$

$$d_k = \frac{1}{\sqrt{\lambda_k}},$$

где λ_k — собственные значения матрицы $\mathbf{A}$.

Все значения λ_k полагаются положительными, в противном случае квадратичная функция не имеет конечного минимума. Формула (5.3.3) свидетельствует о том, что линии уровня функции f в координатах $z^{(k)}$ представляют собой эллипсоиды. Направления их осей совпадают с $z^{(i)}$, а длины осей пропорциональны $\frac{1}{\sqrt{\lambda_k}}$ (вдоль координат с большими λ_k эллипсоиды больше сплюснуты).

Покоординатный спуск в осях $z^{(k)}$ не вызывает трудностей при различных соотношениях между λ_k. Если при минимизации по каждой координате достигать точного локального минимума, то через m шагов получим точный минимум. На рис. 5.3 иллюстрируется этот факт для случая двух переменных ($\mathbf{x}_2 = \mathbf{x}^*$).

Возвращаясь от вектора $\mathbf{z}$ к вектору $\mathbf{x}$, приходим к выводу о том, что линии уровня квадратичной функции (5.3.1) представляют собой эллипсоиды, произвольным образом ориентированные в пространстве. Направления осей определяются собственными векторами матрицы Гессе, а их размеры пропорциональны $\frac{1}{\sqrt{\lambda_k}}$. Если матрица $\mathbf{A}$ плохо обусловлена, то линии уровня имеют сильно вытянутый характер и покоординатный спуск становится крайне неэффективным в осях $x^{(k)}$ (рис. 5.4).

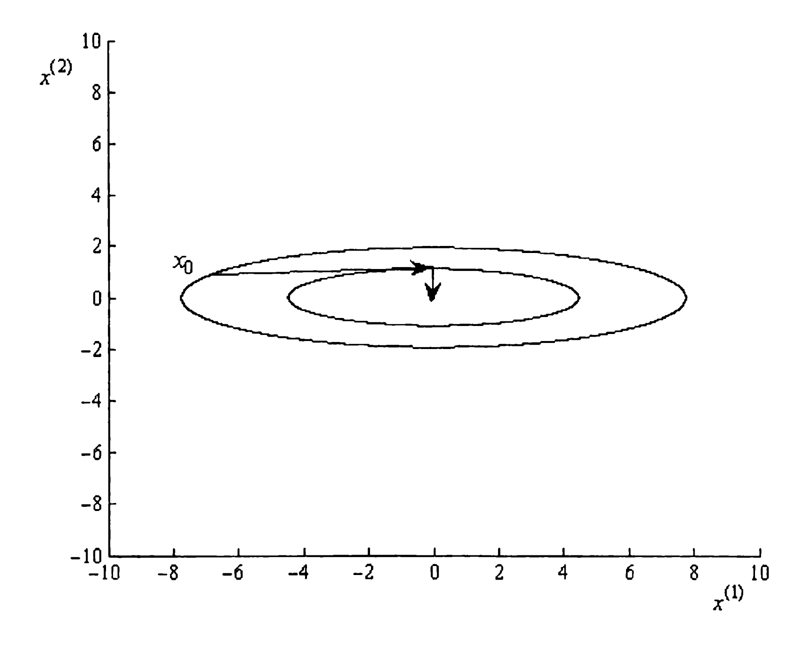


Рис. 5.3. Успешная минимизация вдоль осей эллипса

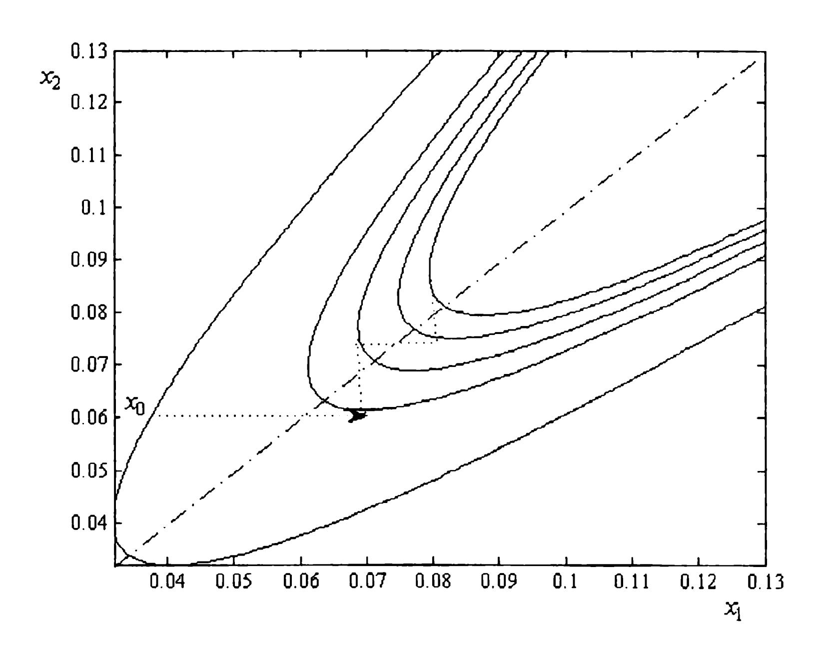


Рис. 5.4. Минимизация овражной функции

Расстояние до точки касания очередной линии уровня оказывается очень малым, и для достижения точки минимума нужно выполнить большое число шагов, которое катастрофически возрастает с увеличением числа обусловленности матрицы Гессе. Как видно из реального примера на рис. 5.4, уже первые шаги метода имеют величину порядка одной сотой, в дальнейшем они еще более уменьшаются, а минимуму функции отвечают значения параметров $x^{(1)} = x^{(2)} = 3$.

Такие функции с сильно вытянутыми линиями уровня и плохо обусловленными матрицами Гессе получили название *овражных* функций. Примером их в трехмерном случае будут следующие два варианта:

- $\lambda_1 \geq \lambda_2 \gg \lambda_3$ — эллипсоид похож на иглу;
- $\lambda_1 \gg \lambda_2 \geq \lambda_3$ — эллипсоид похож на диск.

Ситуация с большим количеством шагов и часто неприемлемым объемом вычислений при минимизации овражных функций характерна и для методов градиентного спуска.

Наконец, минимизация функции (5.3.1) методом Ньютона (5.2.8) с учетом выражений (5.3.2) приводит к следующему результату:

$$\mathbf{x}_{n+1} = \mathbf{x}_n - \left(\frac{\partial^2 f}{\partial \mathbf{x}^2}\right)^{-1} \mathbf{grad}\, f(\mathbf{x}) = \mathbf{x}_n - \mathbf{A}^{-1}\left(\mathbf{A}\mathbf{x}_n + \mathbf{b}\right) = -\mathbf{A}^{-1}\mathbf{b} = \mathbf{x}^*.$$

Таким образом, для квадратичной функции метод Ньютона теоретически приводит в точку минимума за один шаг. Однако на практике для овражной функции применение метода Ньютона связано с обращением плохо обусловленной матрицы $\mathbf{A}$, что влечет за собой хорошо известные последствия в виде потери точности решения.

Выполненные выше преобразования и вид рис. 5.3 и 5.4 подсказывают выход из создавшейся ситуации: вычислить собственные векторы матрицы $\mathbf{A}$, перейти от координат вектора $\mathbf{x}$ к координатам вектора $\mathbf{z}$ и в этих осях применить обычный покоординатный спуск, что, как уже отмечалось, позволяет легко достичь минимума даже при большом разбросе собственных чисел. Такой метод получил название *метода обобщенного покоординатного спуска* [54]. Если минимизируемая функция отлична от квадратичной, то определяют в некоторой точке матрицу Гессе и ее собственные векторы, а затем минимизируют исходную функцию вдоль этих собственных векторов, пока не произойдет значительное замедление сходимости. Затем матрицу Гессе вычисляют в новой точке и т. д. Для нахождения компонентов вектора гради-

ента или элементов матрицы Гессе, когда отсутствуют их явные аналитические выражения, можно использовать формулы численного дифференцирования.

Рассмотрим еще один подход к минимизации функций, в том числе и овражных. Уже отмечалась глубокая связь задачи минимизации функций с задачей решения систем нелинейных уравнений. Такая же связь существует и с задачей решения систем дифференциальных уравнений специального вида.

Пусть все компоненты вектора $\mathbf{x}$ зависят от некоторой переменной τ и являются решением следующей системы дифференциальных уравнений:

$$\frac{d\mathbf{x}(\tau)}{d\tau} = -\mathbf{grad}\, f(\mathbf{x}). \tag{5.3.4}$$

Как при этом меняется функция $f(\mathbf{x})$ с изменением τ? Для ее производной имеем выражение:

$$\frac{df(\mathbf{x})}{d\tau} = \sum_{k=1}^{m} \frac{\partial f}{\partial x^{(k)}} \frac{dx^{(k)}}{d\tau} = \left(\mathbf{grad}\, f(\mathbf{x}), \frac{d\mathbf{x}}{d\tau}\right) = -\left(\mathbf{grad}\, f(\mathbf{x}),\ \mathbf{grad}\, f(\mathbf{x})\right) < 0,$$

свидетельствующее о том, что с ростом τ эта функция непрерывно убывает и достигает минимума в точке, отвечающей нулю ее градиента. Система (5.3.4) называется *дифференциальным уравнением линии спуска*.

Теперь можно "забыть" о задаче минимизации $f(\mathbf{x})$ и решать уравнение (5.3.4) одним из численных методов, рассмотренных в *главе 4*. Компоненты вектора $\mathbf{x}$ будут изменяться в зависимости от τ, одновременно обеспечивая уменьшение $f(\mathbf{x})$.

А если $f(\mathbf{x})$ проявляет овражные свойства?

Для квадратичной функции с учетом (5.3.2) уравнение линии спуска (5.3.4) приобретает вид:

$$\frac{dx(\tau)}{d\tau} = -\mathbf{A}\mathbf{x} - \mathbf{b}. \tag{5.3.5}$$

Матрица $\mathbf{A}$ в рассмотренном "тестовом" случае имеет положительные λ_k и в случае овражности плохо обусловлена. Тогда система (5.3.5) обладает асимптотически устойчивым решением, но при этом является жесткой. Ситуация не меняется и для неквадратичной $f(\mathbf{x})$: *овражная* функция приводит к *жесткости* дифференциального уравнения линии спуска (5.3.4). И хотя для решения последнего приемлемы уже далеко не все алгоритмы *главы 4* (несостоятельны явные методы Рунге — Кутты, Адамса и пр.), с успехом мо-

гут быть использованы специальные методы интегрирования жестких систем, в частности неявные разностные схемы ломаных Эйлера и трапеций.

Таким образом, все сказанное позволяет свести проблему минимизации овражной функции к интегрированию жесткого уравнения (5.3.4). И здесь каждому методу решения жестких систем будет соответствовать свой алгоритм минимизации. Проиллюстрируем сказанное следующими рассуждениями.

Пусть функция $f(\mathbf{x})$ является овражной, а уравнение (5.3.4) — жестким. Попробуем решать его явным методом ломаных Эйлера, понимая полную бесперспективность такого подхода. Тогда формула метода принимает следующий вид:

$$\mathbf{x}_{n+1} = \mathbf{x}_n - h\,\mathbf{grad}\,f(\mathbf{x}_n). \tag{5.3.6}$$

Дальнейшее нетрудно предсказать. Условие устойчивости явного метода ломаных Эйлера (4.3.2) вынуждает использовать крайне малую величину шага h, что в конечном счете приводит к неоправданно большому объему вычислений. Но формула (5.3.6) в точности совпадает с выражением (5.2.5), определяющим метод градиентного спуска минимизации функций. Таким образом, минимизация овражной функции градиентным методом столь же "интеллектуальное" занятие, что и решение жесткой системы явным методом ломаных Эйлера!

Обобщая полученные результаты и результаты, представленные в третьем и пятом разделах, приходим к выводу о том, что в трех весьма далеких, на первый взгляд, задачах плохая обусловленность матрицы порождает три проблемы:

- *плохую обусловленность* при решении линейных систем алгебраических уравнений;
- явление *жесткости* при интегрировании дифференциальных уравнений;
- *овражность* при минимизации функций.

В первом случае это приводит к низкой точности решения, а в двух других — к большому объему вычислительных затрат при использовании традиционных алгоритмов.

Теперь остановимся на некоторых типичных затруднениях (помимо овражности), встречающихся при минимизации функций, и в качестве примера рассмотрим два из них.

1. Функция $f(\mathbf{x})$ имеет существенно различную чувствительность к изменению различных независимых переменных, что может быть следствием

плохого масштабирования. Иллюстрацией может служить овражная функция

$$f\left(x^{(1)},x^{(2)}\right)=100x^{(1)}x^{(1)}+0.01x^{(2)}x^{(2)}, \qquad (5.3.7)$$

сильно зависящая от изменения $x^{(1)}$ и слабо — от $x^{(2)}$, если обе переменные принимают значения одного порядка. Замена переменных с масштабными множителями $x^{(1)}=0.1y^{(1)}$, $x^{(2)}=10y^{(2)}$ устраняет этот недостаток и в новых координатах делает функцию неовражной. В более сложных ситуациях не всегда удается так легко изменить масштаб функций, как в этом примере, особенно если аналитические формулы для $f(\mathbf{x})$ отсутствуют. Однако в любом случае перед началом процесса минимизации рекомендуется предварительно выполнить масштабирование, чтобы чувствительность $f(\mathbf{x})$ ко всем компонентам вектора **x** стала бы соизмеримой.

Здесь уместно отметить, что при интегрировании линейных жестких систем $\mathbf{x}'=\mathbf{A}\mathbf{x}$ такая замена $\mathbf{x}=\mathbf{D}\mathbf{y}$ ничего не давала, поскольку в новых координатах матрица системы оказывалась подобной исходной и имела те же собственные значения:

$$\mathbf{y}'=\left(\mathbf{D}^{-1}\mathbf{A}\mathbf{D}\right)\mathbf{y}.$$

В задаче же минимизации новая матрица Гессе уже не является подобной старой и свойства овражности после замены переменных могут существенно различаться:

$$f=\frac{1}{2}(\mathbf{A}\mathbf{x},\mathbf{x})=\frac{1}{2}(\mathbf{A}\mathbf{D}\mathbf{y},\mathbf{D}\mathbf{y})=\frac{1}{2}\left(\mathbf{D}^{T}\mathbf{A}\mathbf{D}\mathbf{y},\mathbf{y}\right).$$

2. В выражении для $f(\mathbf{x})$ переменные могут оказаться взаимосвязанными, что иллюстрируется простым примером:

$$f\left(x^{(1)},x^{(2)}\right)=\left(x^{(1)}x^{(2)}-6\right)^{2}+\frac{\pi}{2}. \qquad (5.3.8)$$

Минимальное значение $f(\mathbf{x})$ равно $\pi/2$, но в оптимальной точке каждая из переменных $x^{(1)}$ и $x^{(2)}$ может принимать различные значения при заданном произведении $x^{(1)}x^{(2)}=6$. В этих условиях минимум f определяется однозначно не для компонентов $x^{(k)}$, а для некоторых их комбинаций, называемых "комплексами" (или "агрегатами"), число которых меньше, чем количество $x^{(k)}$. Однозначно в координатах $x^{(k)}$ задача может быть решена лишь с привлечением какой-либо дополнительной информации о $x^{(k)}$.

В тех случаях, когда аналитические формулы для $f(\mathbf{x})$ отсутствуют и комплексы явно не определяются, важно предварительно выяснить:

- какие параметры образуют комплексы;
- каков *минимальный объем* дополнительной информации для однозначного определения всех $x^{(k)}$?

Ответить на эти вопросы помогает следующая методика.

Первоначально, минимизируя $f(\mathbf{x})$, находим вектор $\mathbf{x}^*$, обеспечивающий минимум $f(\mathbf{x}^*)=f^*$. Если имеет место образование комплексов, то значению f^* отвечает некоторое множество параметров $\mathbf{x}$, отражающих эти комплексы. Для выделения параметров, входящих в комплексы, изменив один из параметров $x^{(k)}$ от найденной точки $\mathbf{x}^*$ на некоторую величину и зафиксировав его значение, проводят минимизацию за счет остальных параметров. Если в некоторой точке $\mathbf{x}^{**}$ удается получить $f\left(\mathbf{x}^{**}\right)\approx f^*$, то зафиксированный параметр входит в комплекс. Более того, в комплексы входят также все параметры, значения которых в векторах $\mathbf{x}^*$ и $\mathbf{x}^{**}$ различаются.

После этого изменяют и фиксируют поочередно остальные компоненты вектора $\mathbf{x}$, вхождение которых в комплексы на предыдущих шагах не было установлено, и проводят поиск минимума f по оставшимся нефиксированным параметрам.

Если при выполнении этих процедур оказывается невозможным получить значение минимума f, близкое к f^*, то это означает, что фиксированный параметр в комплексы не входит. Тогда его значение в векторе $\mathbf{x}^*$ считается найденным однозначно, фиксируется таковым и на дальнейших этапах не изменяется.

Когда, таким образом, определен перечень параметров, входящих в комплексы, на основании дополнительной априорной информации о процессе число независимо варьируемых $x^{(k)}$ понижают на единицу и повторяют указанную процедуру в пространстве меньшей размерности. Понижение размерности пространства на единицу может осуществляться различными способами, например:

- фиксацией одного из параметров;
- наложением некоторой связи на $x^{(k)}$, в соответствии с которой какой-либо из них будет изменяться в зависимости от другого и т. п.

Вся методика легко автоматизируется и может включать лишь элементы диалога для введения исследователем априорной информации о параметрах модели.

В настоящем разделе мы ограничились рассмотрением задачи безусловной минимизации. Условная минимизация требует специальных подходов, чаще значительно более сложных. В качестве примера обратимся лишь к иллюстрации основной идеи методов штрафных и барьерных функций, позволяющих преобразовать исходную задачу в эквивалентную последовательность задач без ограничений.

Требуется найти минимум функции $f(\mathbf{x})$ при ограничениях на вектор $\mathbf{x}$, записанных в виде неравенств

$$c_k(\mathbf{x}) \geq 0, \qquad k = 1,\ 2,\ \ldots,\ s. \tag{5.3.9}$$

Если безусловная минимизация $f(\mathbf{x})$ приводит к результату, когда все условия (5.3.9) выполняются, то задача мало отличается от предыдущей. Более интересна ситуация, когда минимум достигается на границе разрешенной области, т. е. когда одно или несколько неравенств (5.3.9) одновременно превращаются в равенства $c_k(\mathbf{x}) = 0$.

Вместо $f(\mathbf{x})$ будем несколько раз минимизировать функцию $F(\mathbf{x})$

$$F(\mathbf{x}) = f(\mathbf{x}) + T(\mathbf{x},\alpha), \tag{5.3.10}$$

где $T(\mathbf{x},\alpha)$ — функция, выбираемая специальным образом; α — параметр, принимающий последовательно уменьшающиеся значения, например, $\alpha = 10^{-m}$, m = 1, 2, ... При этом минимум $F(\mathbf{x})$ находят при фиксированном значении α, а полученный результат является начальным приближением для решения задачи при новом α и т. д. В основу выбора $T(\mathbf{x}, \alpha)$ положены следующие соображения.

5.3.1. Метод барьерных функций

Функция $T(\mathbf{x},\alpha)$ называется *барьерной* и выбирается, например, следующим образом:

$$T(\mathbf{x},\alpha) = \alpha \sum_{k=1}^{s} \frac{1}{c_k(\mathbf{x})}. \tag{5.3.11}$$

Начальное приближение $\mathbf{x}_0$ выбирается *внутри* области возможного изменения $\mathbf{x}$, когда все условия (5.3.9) выполняются. В процессе минимизации с

ростом числа итераций n значение $\mathbf{x}_n$ приближается изнутри к границе области, где $c_k(\mathbf{x})=0$, и функция $T(\mathbf{x},\alpha)$ резко возрастает, образуя "барьер". Получившаяся точка минимума используется в качестве начальной для минимизации $T(\mathbf{x},\alpha)$ с меньшим значением α. При этом роль "барьера" сохраняется, но вклад барьерной функции в формирование $F(\mathbf{x})$ уменьшается. $F(\mathbf{x})$ становится ближе к $f(\mathbf{x})$, а новая точка минимума будет еще ближе к границе области. Процесс уменьшения α не должен быть слишком быстрым. Если изначально выбрать значение α очень малым, то существует опасность выйти за границу области при получении очередного приближения $\mathbf{x}_n$ из-за малости $T(\mathbf{x},\alpha)$ везде, кроме границы.

Выбор барьерной функции в форме (5.3.11), разумеется, не является единственно возможным. Так, например, функции барьера с успехом может выполнять и функция следующего вида:

$$T(\mathbf{x},\alpha)=-\alpha\sum_{k=1}^{s}\ln c_k(\mathbf{x}). \qquad (5.3.12)$$

5.3.2. Метод штрафных функций

Здесь начальное приближение $\mathbf{x}_0$ выбирается *вне* области возможного изменения $\mathbf{x}$, когда одно или одновременно несколько условий (5.3.9) нарушаются, т. е. $c_k(\mathbf{x})<0$. Функция $T(\mathbf{x},\alpha)$ в этом случае называется *штрафной* и выбирается, например, следующим образом:

$$T(\mathbf{x},\alpha)=\frac{1}{\alpha}\sum_{k=1}^{s}\left[\min\left(0,c_k(\mathbf{x})\right)\right]^2. \qquad (5.3.13)$$

Параметр α принимает те же значения, что и ранее. Внутри области допустимого изменения $\mathbf{x}$ штрафная функция равна нулю и $F(\mathbf{x})$ совпадает с $f(\mathbf{x})$. При нахождении $\mathbf{x}_n$ вне области к минимизируемой функции $f(\mathbf{x})$ добавляется "штраф" $T(\mathbf{x},\alpha)$, значение которого возрастает по мере уменьшения α. Сказанное приводит к тому, что $\mathbf{x}_n$ извне стремится к границе области. Напомним, что для барьерных функций точка $\mathbf{x}_n$ все время находилась внутри области.

Мы рассмотрели лишь иллюстрацию основной идеи методов штрафных и барьерных функций. Реальные алгоритмы имеют целый ряд весьма важных

подробностей. Так, выбор штрафных и барьерных функций весьма богат, и при построении $F(\mathbf{x})$ целесообразно обеспечить нужную ее гладкость, а также учесть конкретные особенности $f(\mathbf{x})$. Большие возможности по улучшению работы методов скрываются в правиле, по которому будет уменьшаться значение α. Следует учитывать также тот факт, что с уменьшением α функция $F(\mathbf{x})$ становится более овражной и т. д.

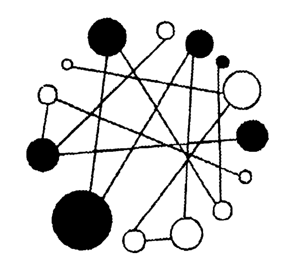
В заключение следует отметить, что в тех случаях, когда ограничения на параметры имеют простой вид, обычная замена переменных позволяет все свести к решению задачи безусловной минимизации. Так, для ограничений $x^{(k)} \geq 0$ или $a \leq x^{(k)} \leq b$ соответствующая замена переменных

$$x^{(k)} = \left(q^{(k)}\right)^2 \text{ и } x^{(k)} = \frac{a+b}{2} + \frac{b-a}{2}\sin q^{(k)}$$

и последующая минимизация без ограничений по $q^{(k)}$ эффективно устраняют возникшие затруднения. Иногда это может привести к появлению дополнительных экстремумов и некоторых других затруднений, однако в большинстве случаев такие замены переменных представляются весьма полезными.

С типичными заменами переменных для различных ограничений, а также с методами минимизации овражных функций можно ознакомиться подробнее по книге [54].

Глава 6



И кое-что еще...

Материал этой главы является более детальным знакомством с проблемами, уже рассматривавшимися в предыдущих главах. В частности, он может изучаться самостоятельно или быть предметом обсуждения на семинарах.

6.1. Сингулярное разложение матрицы и его использование в методе наименьших квадратов

Ранее уже строились два разложения матрицы (LU и QR). Так матрица с различными собственными значениями приводилась к диагональному виду, для решения систем линейных алгебраических уравнений

$$\mathbf{Ax} = \mathbf{b} \qquad (6.1.1)$$

использовалось LU-разложение матрицы, а для поиска собственных значений — QR-разложение. В *главе 2* система (6.1.1) решалась в предположении, что $\det(\mathbf{A}) \neq 0$. А если $\det(\mathbf{A}) = 0$? А если матрица $\mathbf{A}$ прямоугольная? Ответ на эти и многие другие вопросы может быть получен с использованием сингулярного разложения матрицы.

6.1.1. Сингулярное разложение матрицы

Пусть матрица $\mathbf{A}$ в общем случае прямоугольная размера $m \times n$.

Определение. *Сингулярным разложением* матрицы $\mathbf{A}$ с вещественными элементами называется ее представление в виде

$$\mathbf{A} = \mathbf{U\Sigma V}^T, \qquad (6.1.2)$$

$$\mathbf{U} \cdot \mathbf{U}^T = \mathbf{E}_m, \quad \mathbf{V} \cdot \mathbf{V}^T = \mathbf{E}_n,$$

где $\mathbf{U}$ — ортогональная матрица размера $m \times m$; $\mathbf{V}$ — ортогональная матрица размера $n \times n$, а $\mathbf{\Sigma}$ — диагональная матрица размера $m \times n$ с элементами $\sigma_{kj} = 0$ при $k \neq j$ и $\sigma_{kk} = \sigma_k \geq 0$. $\mathbf{E}_m$ и $\mathbf{E}_n$ — единичные матрицы $m \times m$ и $n \times n$ соответственно. Величины σ_k называются *сингулярными числами*, а столбцы матриц $\mathbf{U}$ и $\mathbf{V}$ — *левыми и правыми сингулярными векторами.*

Рассмотрим две симметрические матрицы $\mathbf{A}^T\mathbf{A}$ и $\mathbf{AA}^T$ размера $n \times n$ и $m \times m$ соответственно:

$$\mathbf{A}^T\mathbf{A} = \left(\mathbf{U\Sigma V}^T\right)^T \mathbf{U\Sigma V}^T = \mathbf{V\Sigma}^T\mathbf{U}^T\mathbf{U\Sigma V}^T = \mathbf{V}\left(\mathbf{\Sigma}^T\mathbf{\Sigma}\right)\mathbf{V}^T;$$

$$\mathbf{AA}^T = \mathbf{U\Sigma V}^T\left(\mathbf{U\Sigma V}^T\right)^T = \mathbf{U\Sigma V}^T\mathbf{V\Sigma}^T\mathbf{U}^T = \mathbf{U}\left(\mathbf{\Sigma\Sigma}^T\right)\mathbf{U}^T.$$

Пусть для определенности $m \geq n$. Матрица $\mathbf{A}^T\mathbf{A}$ подобна матрице $\mathbf{\Sigma}^T\mathbf{\Sigma}$ и имеет такие же собственные значения, которые, в свою очередь, равны квадратам сингулярных чисел матрицы $\mathbf{A}$. Аналогично, первые n собственных значений матрицы $\mathbf{AA}^T$ также равны квадратам сингулярных чисел, а остальные $(m-n)$ собственных значений являются нулевыми. Другими словами, сингулярные числа матрицы $\mathbf{A}$ — положительные квадратные корни из ненулевых собственных значений матриц $\mathbf{A}^T\mathbf{A}$ и $\mathbf{AA}^T$

Разложение (6.1.2) допускает неединственность построения. Во-первых, определение допускает любой порядок сингулярных чисел на диагонали матрицы $\mathbf{\Sigma}$. Во-вторых, для кратных сингулярных чисел соответствующие столбцы $\mathbf{U}$ и $\mathbf{V}$ определены неоднозначно. И, наконец, в-третьих, последние $(m-n)$ строк матрицы $\mathbf{\Sigma}$ являются нулевыми, и элементы последних $(m-n)$ столбцов $\mathbf{U}$ могут выбираться произвольно, т. к. вклад в формирование матрицы $\mathbf{A}$ осуществляют лишь первые n столбцов $\mathbf{U}$.

Ранг матрицы $\mathbf{A}$ (максимальное число ее линейно независимых столбцов) совпадает с рангом матрицы $\mathbf{\Sigma}$, т. е. с числом ненулевых сингулярных чисел. Если для $m \geq n$ ранг $\mathbf{A}$ равен n, то такая матрица называется матрицей *пол-*

ного ранга (в противном случае — *недостаточного ранга*). Используется также понятие *эффективного ранга* — количество сингулярных чисел, удовлетворяющих условию $\sigma_k > \varepsilon$, где величина ε характеризует погрешность представления исходных данных. Через сингулярные числа можно выразить число обусловленности матрицы $\mathbf{A}$ полного ранга

$$\text{cond}(\mathbf{A}) = \frac{\sigma_{\max}}{\sigma_{\min}}, \tag{6.1.3}$$

где $\sigma_{\max}$ и $\sigma_{\min}$ — максимальное и минимальное ненулевые сингулярные числа соответственно. Это число обусловленности не совпадает с используемым в (2.1.1), однако обладает похожими свойствами и принимает значения примерно того же порядка.

Построение сингулярного разложения может быть выполнено на основе программы SVD ("singular value decomposition"), имеющей следующие параметры:

```
SVD (NM, M, N, A, W, MATU, U, MATV, V, IERR, RV1)
```

где:

- NM — строчная размерность заявленных двумерных массивов ($NM \geq \max(M, N)$);
- M — число строк матриц $\mathbf{A}$ и $\mathbf{U}$;
- N — число столбцов A и порядок V;
- A — исходная прямоугольная матрица;
- MATU — логическая переменная; ее значение — ".TRUE.", если матрицу U нужно вычислять, и ".FALSE." в противном случае;
- MATV — логическая переменная; ее значение — ".TRUE.", если матрицу V нужно вычислять, и ".FALSE." в противном случае;
- W — содержит N сингулярных чисел матрицы $\mathbf{A}$;
- U — содержит матрицу $\mathbf{U}$ из (6.1.2);
- V — содержит матрицу $\mathbf{V}$ из (6.1.2);
- IERR — значение этой переменной равно нулю, если происходит нормальный выход из программы, и оно равно K, если σ_k не было определено после 30 итераций;
- RV1 — одномерный массив промежуточного хранения.

Алгоритм построения сингулярного разложения содержит два этапа. На основе преобразования Хаусхолдера исходная матрица приводится к двухдиагональному виду $\mathbf{D}_2$

$$\mathbf{A} = \tilde{\mathbf{U}}\mathbf{D}_2\tilde{\mathbf{V}}^T; \quad \mathbf{D}_2 = \tilde{\mathbf{U}}^T\mathbf{A}\tilde{\mathbf{V}},$$

где $\tilde{\mathbf{U}}$ и $\tilde{\mathbf{V}}$ — ортогональные $m \times m$ и $n \times n$ матрицы соответственно; а у матрицы $\mathbf{D}_2$ ненулевые элементы стоят только на диагонали и на первой наддиагонали. Этот этап по своему назначению похож на предварительное приведение матрицы к форме Хессенберга в QR-алгоритме. На втором шаге используется вариант QR-алгоритма, после которого внедиагональные элементы $\mathbf{D}_2$ становятся пренебрежимо малы, а на диагонали выстраиваются сингулярные числа матрицы $\mathbf{A}$.

6.1.2. Метод наименьших квадратов с использованием сингулярного разложения

Обратимся вновь к дискретному случаю среднеквадратичной аппроксимации (другое название — *метод наименьших квадратов* (МНК)). Функция $f(x)$ задается на дискретном множестве m точек следующей таблицей:

x	x_1	x_2	x_3	...	x_m
$f(x)$	$f(x_1)$	$f(x_2)$	$f(x_3)$	...	$f(x_m)$

а ее аппроксимация $Q_{n-1}(x)$ выбирается в виде обобщенного многочлена

$$Q_{n-1}(x) = c_1\varphi_1(x) + c_2\varphi_2(x) + \ldots + c_n\varphi_n(x) = \sum_{k=1}^{n} c_k\varphi_k(x). \quad (6.1.4)$$

Коэффициенты c_k определяются из условия минимума квадрата расстояния ρ^2 между $Q_{n-1}(x)$ и $f(x)$

$$\rho^2 = \left(Q_{n-1}(x) - f(x),\ Q_{n-1}(x) - f(x)\right) = \sum_{i=1}^{m}\left(Q_{n-1}(x_i) - f(x_i)\right)^2 \to \min. \quad (6.1.5)$$

Как уже отмечалось в *разд. 1.13.1*, необходимое условие экстремума

$$\frac{\partial\rho^2(c_k)}{\partial c_k} = 0,\ k = 1,\ 2,\ \ldots,\ n$$

приводит к системе линейных алгебраических уравнений, которая с ростом числа искомых коэффициентов становится катастрофически плохо обусловленной. Наиболее надежным способом возникающие проблемы могут быть решены с использованием сингулярного разложения матрицы. С этой целью несколько переформулируем задачу. Попытка максимально удачно описать табличную функцию с помощью $Q_{n-1}(x)$ приводит к следующим приближенным равенствам:

$$\mathbf{Ac} \approx \mathbf{f};$$
$$\mathbf{c} = \left(c_1, c_2, \ldots, c_n\right)^T; \qquad \mathbf{f} = \left(f(x_1), f(x_2), \ldots, f(x_m)\right)^T, \tag{6.1.6}$$

где элементы прямоугольной $m \times n$ матрицы $\mathbf{A}$ определяются по формулам

$$a_{ik} = \varphi_k(x_i); \quad i = 1,\ 2,\ \ldots,\ m; \quad k = 1,\ 2,\ \ldots,\ n.$$

Введем *вектор невязки* $\mathbf{r} = \mathbf{Ac} - \mathbf{f}$. Задача минимизация его длины

$$(\mathbf{r}, \mathbf{r}) \to \min \tag{6.1.7}$$

полностью совпадает с задачей минимизации величины ρ^2 в (6.1.5). Таким образом, будем искать такой вектор коэффициентов $\mathbf{c}$ в системе (6.1.6), чтобы квадрат длины вектора невязки был бы минимален. Заменяя в (6.1.6) матрицу $\mathbf{A}$ ее сингулярным разложением $\mathbf{U\Sigma V}^T$ и умножая получившиеся уравнения на $\mathbf{U}^T$

$$\mathbf{\Sigma V}^T \mathbf{c} \approx \mathbf{U}^T \mathbf{f},$$

перейдем к новым переменным

$$\tilde{\mathbf{c}} = \mathbf{V}^T \mathbf{c}; \quad \tilde{\mathbf{f}} = \mathbf{U}^T \mathbf{f}; \quad \mathbf{c} = \mathbf{V}\tilde{\mathbf{c}}; \quad \mathbf{f} = \mathbf{U}\tilde{\mathbf{f}}. \tag{6.1.8}$$

С учетом этих обозначений система (6.1.6) примет следующий вид

$$\mathbf{\Sigma}\tilde{\mathbf{c}} \approx \tilde{\mathbf{f}}. \tag{6.1.9}$$

Теперь вектор невязки для (6.1.9) принимает следующий вид $\tilde{\mathbf{r}} = \mathbf{\Sigma}\tilde{\mathbf{c}} - \tilde{\mathbf{f}}$. Так как замены переменных (6.1.8) выполнялись с ортогональными матрицами, длина вектора $\tilde{\mathbf{r}}$ по сравнению с $\mathbf{r}$ осталась прежней. Действительно,

$$(\mathbf{r}, \mathbf{r}) = (\mathbf{Ac} - \mathbf{f}, \mathbf{Ac} - \mathbf{f}) = \left(\mathbf{U\Sigma V}^T \mathbf{c} - \mathbf{f}, \mathbf{U\Sigma V}^T \mathbf{c} - \mathbf{f}\right) =$$
$$= \left(\mathbf{\Sigma V}^T \mathbf{c} - \mathbf{U}^T \mathbf{f}, \mathbf{\Sigma V}^T \mathbf{c} - \mathbf{U}^T \mathbf{f}\right) = \left(\mathbf{\Sigma}\tilde{\mathbf{c}} - \tilde{\mathbf{f}}, \mathbf{\Sigma}\tilde{\mathbf{c}} - \tilde{\mathbf{f}}\right) = (\tilde{\mathbf{r}}, \tilde{\mathbf{r}}).$$

Рассмотрим подробнее решение (6.1.9) для различных сочетаний m и n.

Случай 1. $m \geq n$. Система (6.1.9) разбивается на две подсистемы:

$$\tilde{c}_k \sigma_k \approx \tilde{f}_k,\ k = 1,\ 2,\ \ldots,\ n; \tag{6.1.10}$$

$$0 \approx \tilde{f}_k,\ k = n+1,\ \ldots,\ m. \tag{6.1.11}$$

Здесь σ_k — по-прежнему сингулярные числа. Для удобства будем полагать, что все они упорядочены по убыванию: $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_n \geq 0$.

Для $m = n$ подсистема (6.1.11) отсутствует. Минимум значения ρ^2 и условие (6.1.7) будут иметь место, если система (6.1.10) решается точно (в предположении $\sigma_k \neq 0$)

$$\tilde{c}_k \approx \frac{\tilde{f}_k}{\sigma_k},\ k = 1,\ 2,\ \ldots,\ n. \tag{6.1.12}$$

Значение же минимума ρ^2 полностью определяется системой (6.1.11)

$$(\mathbf{r}, \mathbf{r}) = (\tilde{\mathbf{r}}, \tilde{\mathbf{r}}) = \sum_{k=n+1}^{m} \tilde{f}_k^2 \tag{6.1.13}$$

и для $m = n$ равно нулю.

Теперь обратимся к ситуации с нулевыми значениями σ_k. Первоначально пусть только $\sigma_n = 0$. Тогда последнее уравнение подсистемы (6.1.10) переходит в (6.1.11), увеличивая значение минимума ρ^2, которое, в свою очередь, определяется неоднозначно и остается неизменным при произвольном значении $\tilde{c}_n$. Переходя к исходному вектору коэффициентов **c**, с учетом (6.1.8) имеем

$$\mathbf{c} = \mathbf{V}\tilde{\mathbf{c}} = \mathbf{c}_{\text{opt}} + \tilde{\mathbf{c}}_n \mathbf{v}_n, \tag{6.1.14}$$

где $\mathbf{v}_n$ — последний столбец матрицы **V**, вектор $\mathbf{c}_{\text{opt}}$ определяется однозначно в соответствии с (6.1.12), а значение $\tilde{c}_n$ может задаваться произвольным. Наиболее популярным является задание $\tilde{c}_n = 0$. Так как векторы **c** и $\tilde{\mathbf{c}}$ имеют одинаковую длину, то нулевое значение $\tilde{c}_n$ из всего множества решений выделяет вектор коэффициентов **c** с *минимальной длиной*. В качестве альтернативы можно потребовать, например, чтобы из всего множества решений предпочтение отдавалось вектору **c**, который максимально близок к некоторому вектору $\mathbf{c}^*$. В этом случае $\tilde{c}_n$ выбирается из условия

$$\left(\mathbf{c} - \mathbf{c}^*, \mathbf{c} - \mathbf{c}^*\right) = \left(\mathbf{c}_{\text{opt}} + \tilde{c}_n \mathbf{v}_n - \mathbf{c}^*,\ \mathbf{c}_{\text{opt}} + \tilde{c}_n \mathbf{v}_n - \mathbf{c}^*\right) \to \min.$$

В частности, для $\mathbf{c}^* = 0$ этому условию отвечает $\tilde{c}_n = 0$.

Если нулевыми являются несколько последних сингулярных чисел

$$\sigma_k = 0, \; k = n1, \; n1+1, \; ..., \; n,$$

то множество решений получается еще более многообразным, чем (6.1.14)

$$\mathbf{c} = \mathbf{V}\tilde{\mathbf{c}} = \mathbf{c}_{opt} + \sum_{k=n1}^{n} \tilde{c}_k \mathbf{v}_k. \tag{6.1.15}$$

Здесь $\mathbf{v}_k$ — последние столбцы матрицы $\mathbf{V}$, а произвольно могут выбираться уже несколько последних коэффициентов $\tilde{c}_k$. Как и ранее, их нулевые значения являются наиболее популярным выбором.

Мощным инструментарием служит использование в задаче среднеквадратичной аппроксимации матриц эффективного ранга (количество сингулярных чисел, удовлетворяющих условию $\sigma_k > \varepsilon$, где величина ε характеризует погрешность представления исходных данных). Если матрица $\mathbf{A}$ плохо обусловлена, и число обусловленности (6.1.3) велико, то чувствительность решения к вариации исходных данных оказывается весьма высокой. При этом среди сингулярных чисел есть несколько сравнительно малых, и соответствующие компоненты вектора $\tilde{\mathbf{c}}$, получаемые из (6.1.12), принимают большие значения. Для повышения надежности решения вводится параметр ε, характеризующий точность исходных данных (погрешность эксперимента) или длину разрядной сетки компьютера при записи чисел с плавающей точкой. Все сингулярные числа, меньшие, чем ε, считаются приближенно нулевыми, последние уравнения из (6.1.10) переносятся в (6.1.11), а соответствующим коэффициентам $\tilde{c}_k$ можно задавать произвольные значения. В итоге число обусловленности уменьшается до величины

$$\text{cond}(\mathbf{A}) = \frac{\sigma_{\max}}{\varepsilon}.$$

Тем самым повышается надежность решения задачи. Одновременно значительно уменьшаются по модулю величины коэффициентов $\tilde{c}_k$. Платой за это является некоторое увеличение длины вектора невязки (6.1.13), т. е. значения ρ^2, по сравнению с минимальным значением.

На практике, таким образом, может быть реализован следующий компьютерный диалог. Решающий задачу последовательно увеличивает значение ε.

7 Зак. 3418

При этом повышается надежность получения решения ценой некоторого отклонения ρ^2 от оптимального значения. Согласуя величины ε и ρ^2 с погрешностью исходных данных и требованиями к точности аппроксимации, приходят к разумному для данной задачи компромиссу.

Случай 2. $m < n$. Если все сингулярные числа σ_k ненулевые, то система (6.1.11) отсутствует, а система (6.1.10) принимает вид

$$\tilde{c}_k \sigma_k \approx \tilde{f}_k, \; k = 1, 2, ..., m. \qquad (6.1.16)$$

Первые m значений $\tilde{c}_k$ выбираются из условия

$$\tilde{c}_k \approx \frac{\tilde{f}_k}{\sigma_k}, \; k = 1, 2, ..., m, \qquad (6.1.17)$$

обеспечивая тем самым нулевое значение ρ^2 (длины вектора невязки), а остальные $\tilde{c}_k$ могут выбираться произвольно. Как уже отмечалось, наиболее популярный их выбор $\tilde{c}_k = 0$, что отвечает минимальной длине векторов $\mathbf{c}$ и $\tilde{\mathbf{c}}$, но это не является строго обязательным. В общем случае при переходе от вектора $\tilde{\mathbf{c}}$ к вектору $\mathbf{c}$ используется формула, аналогичная (6.1.15):

$$\mathbf{c} = \mathbf{V} \cdot \tilde{\mathbf{c}} = \mathbf{c}_{\text{opt}} + \sum_{k=m+1}^{n} \tilde{c}_k \mathbf{v}_k. \qquad (6.1.18)$$

Если некоторые σ_k равны нулю, то последние уравнения (6.1.16) принимают вид (6.1.11), образуя, как и ранее, ненулевое минимальное значение ρ^2, а количество свободно задаваемых значений $\tilde{c}_k$ в (6.1.18) увеличивается на число нулевых σ_k.

Приведенные ранее рассуждения, включая вид решения в форме (6.1.15) или (6.1.18), получили название *сингулярный анализ*.

В качестве иллюстрации изложенного обратимся к весьма яркому, приведенному в книге [14] примеру обработки данных Бюро переписи о населении США с 1900 по 1970 годы.

Год	1900	1910	1920	1930	1940	1950	1960	1970
Население	75994575	91972266	105710620	123203000	131669275	150697361	179323175	203211926

Эти данные аппроксимировались полиномом второй степени

$$Q_2(t) = c_1 + c_2 t + c_3 t^2. \qquad (6.1.19)$$

Применение среднеквадратичной аппроксимации без использования сингулярного разложения с обычной точностью представления данных порядка 7 десятичных разрядов дало следующие результаты:

$$c_1 \approx -0.372 \cdot 10^5, \quad c_2 \approx 0.368 \cdot 10^2, \quad c_3 \approx -0.905 \cdot 10^{-2},$$

а для двойной точности представления порядка 15 десятичных разрядов результаты оказались следующими:

$$c_1 \approx 0.373 \cdot 10^5, \quad c_2 \approx -0.402 \cdot 10^2, \quad c_3 \approx 0.108 \cdot 10^{-1}.$$

Несовпадение имеет место даже по знаку.

Теперь используем сингулярное разложение. Матрица **A** системы (6.1.6) в соответствии с исходной таблицей имеет размер 8×3 и следующие сингулярные числа:

$$\sigma_1 \approx 0.106 \cdot 10^8, \quad \sigma_2 \approx 0.648 \cdot 10^2, \quad \sigma_1 \approx 0.346 \cdot 10^{-3}.$$

Ее недопустимо большое число обусловленности

$$\mathrm{cond}(\mathbf{A}) = \frac{\sigma_{\max}}{\sigma_{\min}} = \frac{\sigma_1}{\sigma_3} \approx 0.306 \cdot 10^{11}$$

полностью объясняет приведенные результаты. Используя матрицу с эффективным рангом равным двум (например, для значения $\varepsilon \approx 0.6 \cdot 10^2$), т. е. приближенно считая $\sigma_3 \approx 0$, значительно уменьшаем число обусловленности до

$$\mathrm{cond}(\mathbf{A}) \approx \frac{\sigma_1}{\varepsilon} \approx 0.177 \cdot 10^6$$

и обеспечиваем не только очень хорошее совпадение результатов:

- $c_1 \approx -0.166 \cdot 10^{-2}$, $c_2 \approx -0.162 \cdot 10^1$, $c_3 \approx 0.869 \cdot 10^{-3}$ — обычная точность;
- $c_1 \approx -0.167 \cdot 10^{-2}$, $c_2 \approx -0.162 \cdot 10^1$, $c_3 \approx 0.871 \cdot 10^{-3}$ — двойная точность,

но и ожидаемые много меньшие значения коэффициентов c_k.

Приведенный пример убеждает в эффективности использования сингулярного разложения в подобных случаях. Однако исходная задача могла быть решена проще и без столь мощного инструментария, если внимательно была бы проанализирована причина плохой обусловленности ее решения. Основой всех неприятностей является тот факт, что значения элементов первой строки приведенной таблицы относительно близки.

Если вместо аппроксимации (6.1.19) использовать следующий полином

$$Q_2(t) = c_1 + c_2(t-1900) + c_3(t-1900)^2,$$

то интервал [1900, 1970] превращается в интервал [0, 70], а число обусловленности становится равным $\operatorname{cond}(\mathbf{A}) \approx 0.575 \cdot 10^4$. Еще более удачная аппроксимация

$$Q_2(t) = c_1 + c_2\left(\frac{t-1935}{10}\right) + c_3\left(\frac{t-1935}{10}\right)^2$$

приводит к интервалу $[-3.5,\ 3.5]$ и числу обусловленности $\operatorname{cond}(\mathbf{A}) \approx 10.7$. В использовании сингулярного разложения в обоих случаях нет никакой необходимости. При этом полезно заметить, что последние аппроксимации остались в классе полиномов второй степени.

6.1.3. Псевдообратная матрица

При анализе систем линейных алгебраических уравнений

$$\mathbf{Ax} = \mathbf{b} \tag{6.1.20}$$

в *главе 2* предполагалось, что матрица $\mathbf{A}$ — квадратная и неособенная, а решение в таком случае однозначно записывается через обратную матрицу

$$\mathbf{x} = \mathbf{A}^{-1}\mathbf{b}. \tag{6.1.21}$$

А, если $\det(\mathbf{A}) = 0$ или матрица $\mathbf{A}$ прямоугольная, что понимается под решением задачи (6.1.20)? Можно ли записать решение в форме, похожей на (6.1.21), и что в таком случае является аналогом обратной матрицы?

В общем случае под решением (6.1.20) понимается вектор $\mathbf{x}$, минимизирующий квадрат длины вектора невязки

$$(\mathbf{r},\mathbf{r}) = (\mathbf{Ax}-\mathbf{b}, \mathbf{Ax}-\mathbf{b}) \to \min.$$

Если задача решается неоднозначно, то из всего множества решений выбирается то, которое обладает наименьшей длиной $(\|\mathbf{x}\| \to \min)$. При этом решение записывается в виде

$$\mathbf{x} = \mathbf{A}^{+}\mathbf{b}, \tag{6.1.22}$$

где под $\mathbf{A}^{+}$ понимается *псевдообратная матрица*. Если матрица $\mathbf{A}$ имеет полный столбцовый ранг, то $\mathbf{A}^{+} = \left(\mathbf{A}^T\mathbf{A}\right)^{-1}\mathbf{A}^T$.

Действительно, для вектора невязки имеем

$$(\mathbf{r},\mathbf{r}) = (\mathbf{Ax}-\mathbf{b},\mathbf{Ax}-\mathbf{b}) = (\mathbf{Ax},\mathbf{Ax}) - 2(\mathbf{Ax},\mathbf{b}) + (\mathbf{b},\mathbf{b}) =$$
$$= \left(\mathbf{A}^T\mathbf{Ax},\mathbf{x}\right) - 2\left(\mathbf{x},\mathbf{A}^T\mathbf{b}\right) + (\mathbf{b},\mathbf{b}) \to \min.$$

Необходимое условие минимума порождает следующий результат:

$$\frac{\partial(\mathbf{r},\mathbf{r})}{\mathbf{x}} = 2\mathbf{A}^T\mathbf{A}x - 2\mathbf{A}^T\mathbf{b} = \mathbf{0},$$

$$\mathbf{x} = \left(\mathbf{A}^T\mathbf{A}\right)^{-1}\mathbf{A}^T\mathbf{b} = \mathbf{A}^+\mathbf{b}; \quad \mathbf{A}^+ = \left(\mathbf{A}^T\mathbf{A}\right)^{-1}\mathbf{A}^T.$$

В общем случае псевдообратная матрица может быть определена следующим образом.

Определение. *Псевдообратной матрицей* Мура—Пенроуза, называется матрица $\mathbf{X} = \mathbf{A}^+$, удовлетворяющая следующим четырем условиям:

- $\mathbf{AXA} = \mathbf{A}$;
- $\mathbf{XAX} = \mathbf{X}$;
- $\mathbf{AX}$ — симметрическая;
- $\mathbf{XA}$ — симметрическая.

Можно показать, что такая матрица $\mathbf{X}$ существует и единственна, и для ее построения применяется сингулярное разложение. С этой целью для заданного числа σ определим число σ^+:

$$\sigma^+ = \begin{cases} \dfrac{1}{\sigma}, & \text{если } \sigma \neq 0; \\ 0, & \text{если } \sigma = 0. \end{cases} \qquad (6.1.23)$$

Если для матрицы $\mathbf{A} = \sigma$ размера 1×1 определить $\mathbf{X}$ как $\mathbf{X} = \sigma^+$, то все четыре условия очевидно выполняются. Теперь в общем случае для матрицы $\mathbf{\Sigma}$ размера $m\times n$ введем матрицу $\mathbf{\Sigma}^+$ размера $n\times m$:

$$\mathbf{\Sigma} = \begin{pmatrix} \sigma_1 & 0 & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & \dots & 0 & \sigma_N \\ 0 & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 \end{pmatrix}; \qquad \mathbf{\Sigma}^+ = \begin{pmatrix} \sigma_1^+ & 0 & 0 & \dots & 0 \\ \dots & \dots & 0 & \dots & 0 \\ 0 & \dots & \sigma_N^+ & \dots & 0 \\ & & & & \end{pmatrix}.$$

Здесь иллюстрируется случай $m \geq n$, что не является принципиальным. Тогда для исходной матрицы $\mathbf{A} = \mathbf{U\Sigma V}^T$ ее псевдообратная задается формулой

$$\mathbf{X} = \mathbf{A}^+ = \mathbf{V\Sigma}^T \mathbf{U}^T.$$

Действительно,

$$\mathbf{AXA} = \mathbf{U\Sigma V}^T \mathbf{V\Sigma}^T \mathbf{U}^T \mathbf{U\Sigma V}^T = \mathbf{U\Sigma\Sigma}^T \mathbf{\Sigma V}^T = \mathbf{U\Sigma V}^T = \mathbf{A};$$

$$\mathbf{XAX} = \mathbf{V\Sigma}^T \mathbf{U}^T \mathbf{U\Sigma V}^T \mathbf{V\Sigma}^T \mathbf{U}^T = \mathbf{V\Sigma}^T \mathbf{\Sigma\Sigma}^T \mathbf{U}^T = \mathbf{V\Sigma}^T \mathbf{U}^T = \mathbf{X},$$

а матрицы $\mathbf{AX}$ и $\mathbf{XA}$ очевидно являются симметрическими. Вариант $m < n$ иллюстрируется аналогично. Введем обозначения для элементов всех рассматриваемых матриц:

$$\mathbf{A} \Rightarrow a_{ij}; \quad \mathbf{A}^+ \Rightarrow a_{ij}^+; \quad \mathbf{U} \Rightarrow u_{ij}; \quad \mathbf{V} \Rightarrow v_{ij}.$$

Тогда для a_{ij} и a_{ij}^+ имеем:

$$a_{ij} = \sum_{k=1}^{n} \sigma_k u_{ik} v_{jk}; \qquad a_{ij}^+ = \sum_{\sigma_k \neq 0} \frac{v_{ik} u_{jk}}{\sigma_k}.$$

Как и ранее в методе наименьших квадратов, для уменьшения числа обусловленности и повышения надежности результатов можно ввести *эффективную псевдообратную матрицу* $\mathbf{A}_\varepsilon^+$. С этой целью аналогично (6.1.23) введем число σ_ε^+:

$$\sigma_\varepsilon^+ = \begin{cases} \dfrac{1}{\sigma}, & \text{если } \sigma > \varepsilon; \\ 0, & \text{в противном случае.} \end{cases}$$

Это позволяет задать элементы $\mathbf{A}_\varepsilon^+$ следующим образом:

$$a_{ij}^+(\varepsilon) = \sum_{\sigma_k > \varepsilon} \frac{v_{ik} u_{jk}}{\sigma_k}. \tag{6.1.24}$$

Непосредственной проверкой легко убедиться, что эффективная псевдообратная матрица $\mathbf{X} = \mathbf{A}^+$ удовлетворяет трем последним условиям для $\mathbf{A}^+$:

- $\mathbf{XAX} = \mathbf{X}$;
- $\mathbf{AX}$ — симметрическая;
- $\mathbf{XA}$ — симметрическая,

а первое условие выполняется с точностью до ε

$$\|\mathbf{AXA}-\mathbf{A}\| \le \varepsilon \ .$$

Для ненулевого значения ε матрица $\mathbf{X}$, удовлетворяющая этим четырем условиям, может оказаться неединственной. Однако матрица с элементами (6.1.24) дополнительно обладает наименьшей нормой $\|\mathbf{X}\|$.

6.2. Понятие некорректно поставленной задачи

Среди возникающих на практике задач важное место занимают те, решения которых неустойчивы к малым возмущениям исходных данных. Так как реально точность исходных данных всегда ограничена, это приводит к неединственности решения в рамках заданной погрешности. В таких случаях говорят о *некорректно поставленных задачах*.

Для некоторой количественной задачи введем область "исходных данных" $\mathbf{U}$ с элементами $\mathbf{u} \in \mathbf{U}$ и область "решений" $\mathbf{Z}$ с элементами $\mathbf{z} \in \mathbf{Z}$. При этом решение $\mathbf{z}$ находим по исходным данным $\mathbf{u}$ в соответствии с некоторым правилом $\mathbf{z} = R(\mathbf{u})$. Метрические пространства $\mathbf{U}$ и $\mathbf{Z}$ могут иметь самую различную природу, определяемую постановкой задачи, поэтому для каждого из них введем расстояния между элементами $\rho_U(\mathbf{u}_1,\mathbf{u}_2)$ и $\rho_Z(\mathbf{z}_1,\mathbf{z}_2)$; $\mathbf{u}_1$, $\mathbf{u}_2 \in \mathbf{U}$; $\mathbf{z}_1$, $\mathbf{z}_2 \in \mathbf{Z}$. Пусть каждому элементу $\mathbf{u} \in \mathbf{U}$ отвечает единственное решение $\mathbf{z} = R(\mathbf{u})$; $\mathbf{z} \in \mathbf{Z}$.

Определение. Задача определения решения $\mathbf{z} = R(\mathbf{u})$ из пространства $\mathbf{Z}$ по исходным данным $\mathbf{u} \in \mathbf{U}$ называется *устойчивой* на пространствах ($\mathbf{Z}$, $\mathbf{U}$), если

$$(\forall \varepsilon > 0)(\exists \delta(\varepsilon) > 0)(\rho_U(\mathbf{u}_1, \mathbf{u}_2) \le \delta(\varepsilon) \Rightarrow \rho_Z(\mathbf{z}_1, \mathbf{z}_2) \le \varepsilon),$$

где $z_1 = R(\mathbf{u}_1)$, $z_2 = R(\mathbf{u}_2)$, $\mathbf{u}_1$, $\mathbf{u}_2 \in \mathbf{U}$; $\mathbf{z}_1$, $\mathbf{z}_2 \in \mathbf{Z}$.

Определение. Задача определения решения $\mathbf{z}$ из пространства $\mathbf{Z}$ по исходным данным $\mathbf{u} \in \mathbf{U}$ называется корректно поставленной на метрических пространствах ($\mathbf{Z}$, $\mathbf{U}$), если выполняются следующие два условия:

- для всякого элемента $\mathbf{u} \in \mathbf{U}$ существует определяемое однозначно решение $\mathbf{z} \in \mathbf{Z}$;
- задача устойчива на пространствах ($\mathbf{Z}$, $\mathbf{U}$).

Нарушение любого из этих требований делает задачу *некорректно поставленной.*

Рассмотрим некоторые примеры некорректно поставленных задач. Не очень внимательный читатель будет удивлен тем, что в приведенных далее примерах нет ничего экзотичного. Более того, все рассматриваемые ситуации уже встречались в предыдущих главах. Следует обратить внимание на то, что превратить задачу из некорректно поставленной в корректную удается чаще всего привлечением *дополнительной информации* о решении.

Пример 1. Самым "свежим" примером является материал предыдущего раздела. В задаче (6.1.1) с $\det(\mathbf{A})=0$ или для прямоугольной матрицы оказывается нарушенным первое условие (единственности решения), особенно при введении эффективной псевдообратной матрицы. Справиться с возникающими трудностями удается за счет дополнительной информации о решении. Например, оказывается предпочтение решению с наименьшей нормой.

Пример 2. Возникновение "комплексов" параметров в задачах минимизации функций (*см. разд. 5.3* и формулу (5.3.8)). Опять нарушено первое условие, и проблема разрешается лишь при использовании дополнительной информации о решении.

Пример 3. Решение систем линейных дифференциальных уравнений $(\mathbf{x}'=\mathbf{A}\mathbf{x})$, когда одно или несколько собственных значений матрицы $\mathbf{A}$ имеет положительную вещественную часть $(\mathrm{Re}\lambda_k>0)$. Здесь нарушено второе условие (условие устойчивости). Близость решений для $t\to\infty$ не удается обеспечить даже при очень малом отличии в начальных условиях.

Пример 4. Задача численного дифференцирования. Вновь нарушено условие устойчивости. В *разд. 1.12* уже рассматривался пример двух функций $f(x)$ и $g(x)$

$$f(x);\quad g(x)=f(x)+\frac{1}{N}\sin\left(N^2x\right);$$

$$\frac{df(x)}{dx};\ \frac{dg(x)}{dx}=\frac{df(x)}{dx}+N\cos\left(N^2x\right);$$

когда с ростом величины N обе функции $f(x)$ и $g(x)$ становятся друг к другу все ближе и ближе, а их производные... все дальше и дальше! Таким образом, близость $f(x)$ и $g(x)$ еще не гарантирует близости их производных.

Пример 5. Типичным "поставщиком" проблем некорректности являются всевозможные так называемые *обратные задачи*, когда по заданному реше-

нию $\mathbf{z} \in \mathbf{Z}$ требуется восстановить исходные данные $\mathbf{u} \in \mathbf{U}$, т. е. $\mathbf{u} = R^*(\mathbf{z})$. Здесь весьма популярным примером из практики моделирования различных систем является задача параметрической идентификации, в которой необходимо подобрать параметры модели так, чтобы результаты моделирования были максимально близки к экспериментальным данным, взятым с реального объекта.

Более глубокое знакомство с примерами некорректных задач, методами их решения и многочисленными проблемами, возникающими в этой области, можно реализовать по книге [45].

6.3. Свойства жестких систем дифференциальных уравнений

Введенное в *разд. 4.3* понятие жесткой системы носило описательный и не вполне строгий характер. Наиболее важным моментом было существование двух принципиально различных участков решения: начального пограничного слоя малой продолжительности $\tau_{\text{ПС}}$ с большими производными и значительно большего $(T \gg \tau_{\text{ПС}})$ второго участка со сравнительно малыми производными, где решение носит относительно спокойный характер. Для линейных жестких систем вида (4.3.1) типична ситуация, когда матрица $\mathbf{A}$ плохо обусловлена, а для нелинейных жестких систем

$$\frac{d\mathbf{x}}{dt} = \mathbf{f}(t, \mathbf{x}) \tag{6.3.1}$$

плохо обусловленной часто оказывается матрица Якоби $\dfrac{\partial \mathbf{f}}{\partial \mathbf{x}}$ на решении. Такое описание содержит целый ряд неопределенностей. Что такое "большие" и "малые" производные? Каким должно быть соотношение между производными внутри и вне пограничного слоя, чтобы можно было судить о наличии жесткости? Ответ на эти вопросы не однозначен и определяется решаемой задачей. Имеется полная аналогия, например, с числами обусловленности $\text{cond}(\mathbf{A})$ в (2.1.1) или $k(\mathbf{A})$

$$\text{cond}(\mathbf{A}) = \|\mathbf{A}\| \|\mathbf{A}^{-1}\|; \quad k(\mathbf{A}) = \frac{\max\limits_k |\lambda_k|}{\min\limits_k |\lambda_k|}; \quad \text{cond}(\mathbf{A}) \geq k(\mathbf{A}).$$

Обычно говорят о плохой обусловленности матрицы, если $k(\mathbf{A}) \gg 1$. Однако в каждом конкретном случае разные величины $k(\mathbf{A})$ могут считаться большими.

В предлагаемом далее определении также приходится мириться с такого рода неопределенностью. Малая продолжительность $\tau_{\text{ПС}}$ пограничного слоя по сравнению с длиной полного промежутка наблюдения решения $[a,\ b]$ задается неравенством

$$\tau_{\text{ПС}} \ll b - a, \tag{6.3.2}$$

а значения производных вне пограничного слоя полагаются меньшими, чем значения внутри него, в N раз, где $N \gg 1$.

Линеаризация правой части системы (6.3.1) в окрестности начальной точки $\mathbf{x}_0$

$$\mathbf{f}(t,\mathbf{x}) = \mathbf{f}(t,\mathbf{x}_0) + \frac{\partial \mathbf{f}}{\partial \mathbf{x}}(\mathbf{x}_0) \cdot (\mathbf{x} - \mathbf{x}_0) + \ldots \tag{6.3.3}$$

позволяет убедиться в том, что для возникновения больших производных внутри пограничного слоя матрица Якоби системы (6.3.1) должна иметь большие по модулю собственные значения. Производные компонентов $x^{(k)}(t)$ вектора $\mathbf{x}(t)$

$$\mathbf{x}(t) = \left(x^{(1)}(t), x^{(2)}(t), \ldots, x^{(m)}(t)\right)^T$$

могут достигать величин порядка

$$L \max_{t \in [t_0, t_0 + T]} \left| x^{(k)}(t) \right|,$$

где значение L определяется неравенствами

$$0 < L \le \rho\left(\frac{\partial \mathbf{f}}{\partial \mathbf{x}}\right) \le \left\| \frac{\partial \mathbf{f}}{\partial \mathbf{x}} \right\|, \tag{6.3.4}$$

$\rho\left(\dfrac{\partial \mathbf{f}}{\partial \mathbf{x}}\right)$ — максимальный модуль собственных значений матрицы Якоби (спектральный радиус), $\|\bullet\|$ — принятая норма матрицы. Поэтому вне пограничного слоя потребуем, чтобы значения производных были меньше в N раз.

Важно отметить, что принадлежность системы дифференциальных уравнений (6.3.1) к жестким на промежутке $[a,\ b]$ предполагает проявление характерных свойств жесткости на *любом* отрезке $[t_0,\ t_0+T]$ внутри $[a,\ b]$. Это, в первую очередь, указывает на потенциальную возможность возникновения пограничного слоя, какую бы точку $t_0 \in [a,\ b]$ мы ни выбрали за начальную. Необходимость такого свойства иллюстрирует следующий пример:

$$\frac{dx}{dt} = \alpha(t)\cdot(x-1), \quad t\in[0,1], \quad x(0)=0, \ \ \alpha(t) = -10^5\exp\left(-10^4 t\right)-1. \tag{6.3.5}$$

Решение этого уравнения

$$x(t) = -\exp\left(\int_0^t \alpha(\tau)d\tau\right)+1 = \exp\left(10\cdot\exp\left(-10^4 t\right)-10\right)\cdot\exp(-t)+1$$

внешне очень похоже на решение жесткого уравнения. Есть и начальный участок с большими производными, и последующее "плавное" изменение $x(t)$. Однако величина $\alpha(t)$ остается большой по модулю только в малой окрестности начальной нулевой точки. Уже для любой начальной точки $t_0 > 0.001$ величина $\alpha(t)$ практически равна -1, и на отрезке $[0,\ 1]$ уравнение (6.3.6) не является жестким.

Определение. Система обыкновенных дифференциальных уравнений (6.3.1) называется *жесткой* на отрезке изменения независимой переменной $[a,\ b]$, если при любом векторе начальных значений $\mathbf{x}(t_0)$ и на любом отрезке $[t_0,\ t_0+T]\subset[a,\ b]$ найдутся такие числа $\tau_{ПС}$, L, N, удовлетворяющие (6.3.2) и (6.3.4), что справедливы неравенства

$$\left|\frac{dx^{(k)}}{dt}\right|_{t>t_0+\tau_{ПС}} \le \frac{L}{N}\max_{t\in[t_0,\,t_0+T]}\left|x^{(k)}(t)\right|, \quad k=1,\ 2,\ \ldots,\ m. \tag{6.3.6}$$

$$t_0+\tau_{ПС} \le t \le t_0+T, \qquad N \gg 1.$$

Если начальные условия таковы, что пограничный слой явно присутствует, то величина N дает представление о том, во сколько раз уменьшились производные за пограничным слоем.

Важным моментом этого определения является неразрывная связь понятия жесткости системы (6.3.1) с величиной промежутка наблюдения решения, отражаемая неравенством (6.3.2). Если жесткую на $[a,\ b]$ систему рассмот-

реть на промежутке $[a,\ a+\tau_{ПС}]$, включающем только пограничный слой, то в этом случае ее нельзя признать жесткой, т. к. никакого различия в характере поведения решения не наблюдается. С другой стороны, система линейных уравнений

$$\frac{d\mathbf{x}}{dt} = \mathbf{A}\mathbf{x} + \mathbf{f}(t) \tag{6.3.7}$$

с матрицей второго порядка и собственными значениями $\lambda_1 = -1$, $\lambda_2 = -2$ имеет своими частными решениями две экспоненты с достаточно близкими показателями и не является жесткой на промежутке [0, 1]. Однако, например, на промежутке [0, 100] эта система уже может быть жесткой, если $\mathbf{f}(t)$ — медленно изменяющаяся функция. Здесь различие проявляется между частными решениями, выраженными этими экспонентами, и частным решением неоднородного уравнения, определяемым видом $\mathbf{f}(t)$.

В общем случае поведение решения нелинейных систем может быть очень сложным. Например, система может быть жесткой на одних участках изменения независимой переменной и нежесткой — на других.

Подводя итоги сказанному, сформулируем некоторые свойства жестких систем. Первоначально обратимся к линейным системам с постоянной матрицей

$$\frac{d\mathbf{x}}{dt} = \mathbf{A}\mathbf{x}. \tag{6.3.8}$$

Свойство 1. Для жестких систем "почти всегда" существуют два участка решения с принципиально различным характером поведения его составляющих, причем продолжительность первого участка значительно меньше, чем второго $(\tau_{ПС} \ll b-a)$. "Почти всегда" — потому, что можно подобрать начальные условия с целью полного устранения пограничного слоя, хотя специфика уравнений, естественно, не изменится.

В общем случае пограничный слой может иметь достаточно сложную структуру, если решение в нем, в свою очередь, описывается комбинацией экспонент. Так для системы (6.3.8) с матрицей третьего порядка и собственными значениями

$$\lambda_1 = -10^5, \quad \lambda_1 = -10^3, \quad \lambda_1 = -1; \qquad t \in [0,1]$$

пограничный слой длительностью $\tau_{ПС} \sim 0.003 - 0.005$ неоднороден и, в свою очередь, разделяется на два участка, первый из которых $\tau_{ПС1} \sim 3\cdot 10^{-5}$ значительно меньше второго $(\tau_{ПС1} \ll \tau_{ПС})$.

Свойство 2. Для линейной системы (6.3.8) собственные значения λ_k и собственные векторы матрицы **A** полностью определяют характер частных решений. Поэтому естественно выявить те требования, которым должны удовлетворять λ_k в жесткой системе, считая для простоты, что кратные собственные значения отсутствуют.

Принято считать, что в жесткой системе (6.3.8) матрица **A** обладает большим числом обусловленности, т. е. $k(\mathbf{A}) \gg 1$. На практике это часто имеет место, но не является обязательным. Исключением, например, является рассмотренный случай группы близких собственных значений $(\lambda_1 = -1,\ \lambda_2 = -2)$ и большого промежутка наблюдения решения $(t \in [0,\ 100])$. Из плохой обусловленности **A** жесткость (6.3.8) также следует далеко не всегда. Это нетрудно заметить уже на системе третьего порядка с собственными значениями

$$\lambda_1 = -1, \quad \lambda_{2,3} = -1 \pm 10^4 \cdot i, \quad t \in [0,1].$$

Величина $k(\mathbf{A}) > 10^4$, а решение (6.3.8) не удовлетворяет условию (6.3.6). На всем отрезке [0, 1] наблюдается сильно осциллирующее решение, и никакого противоречия между производными на отдельных участках не наблюдается.

Не выдерживает критики и предложение всегда считать жесткой систему, у которой собственные значения имеют сильно отличающиеся вещественные части $\mathrm{Re}(\lambda_k)$

$$\mu(\mathbf{A}) = \frac{\max\limits_k |\mathrm{Re}(\lambda_k)|}{\min\limits_k |\mathrm{Re}(\lambda_k)|} \gg 1.$$

Это требование также не всегда гарантирует условие (6.3.6) медленного изменения решения вне пограничного слоя, что можно видеть на примере

$$\lambda_1 = -10^4, \quad \lambda_{2,3} = -1 \pm 10^4 \cdot i, \quad t \in [0,1].$$

Кроме того, определяя свойство жесткости, при любом наборе собственных значений нельзя игнорировать длину промежутка наблюдения решения.

Какими же свойствами должны обладать λ_k в жесткой системе?

Введем следующие обозначения:

$$\lambda_k = \alpha_k + i\omega_k, \quad \alpha_k = \mathrm{Re}(\lambda_k), \quad \omega_k = \mathrm{Im}(\lambda_k), \quad i = \sqrt{-1}$$

и запишем решение (6.3.8) через матричную экспоненту

$$\mathbf{x}(t) = e^{\mathbf{A}t} \cdot \mathbf{x}(0).$$

Так как неравенство (6.3.6) должно выполняться для любых начальных условий, выберем $\mathbf{x}(0)$ так, чтобы l-ый компонент вектора решения имел вид:

$$x^{(l)}(t) = C_k \exp(\alpha_k t) \cdot \cos(\omega_k t + \varphi_k). \tag{6.3.9}$$

Пусть первоначально $\alpha_k < 0$. Выбирая $C_k = 1$ и $\varphi_k = 0$, получаем решение с $\max\limits_t \left| x^{(l)}(t) \right| = x^{(l)}(0) = 1$. Дифференцируя (6.3.9), имеем

$$\frac{dx^{(l)}(t)}{dt} = |\lambda_k| \exp(\alpha_k t) \cdot \cos(\omega_k t + \phi_k), \;\; \phi_k = \operatorname{arctg} \frac{\omega_k}{\alpha_k}.$$

Неравенство (6.3.6) при всех $t > \tau_{\text{ПС}}$ требует выполнение условия

$$|\lambda_k| \exp\left(\operatorname{Re}(\lambda_k) \tau_{\text{ПС}}\right) \le \frac{L}{N}, \quad L = \max_k |\lambda_k|, \quad N \gg 1. \tag{6.3.10}$$

Теперь рассмотрим случай $\alpha_k \ge 0$. Выберем начальные условия таким образом, чтобы максимум модуля l-ого компонента решения достигался в точке T. При этом $C_k = \exp\left(-\operatorname{Re}(\lambda_k) T\right)$. Тогда для производной $x^{(l)}(t)$ получим

$$\frac{dx^{(l)}(t)}{dt} = |\lambda_k| \exp\left(\operatorname{Re}(\lambda_k)(t - T)\right) \cdot \cos\left(\omega_k t + \phi_k\right).$$

Для выполнения (6.3.6) при всех $t > \tau_{\text{ПС}}$ необходимо, чтобы

$$|\lambda_k| \le \frac{L}{N}. \tag{6.3.11}$$

Объединяя условия (6.3.10) и (6.3.11), получаем необходимые требования на собственные значения в жесткой системе (6.3.8):

$$|\lambda_k| \exp\left(\operatorname{Re}(\lambda_k) \tau_{\text{ПС}}\right) \le \frac{L}{N}, \text{ если } \operatorname{Re}(\lambda_k) < 0;$$

$$|\lambda_k| \le \frac{L}{N}, \text{ если } \operatorname{Re}(\lambda_k) \ge 0; \tag{6.3.12}$$

$$L = \max_k |\lambda_k|, \; N \gg 1, \; \tau_{\text{ПС}} \ll T.$$

Непосредственно из этих условий следует, что в жесткой системе не может быть больших по модулю собственных значений (порядка L) с положительной вещественной частью. Для собственных значений, имеющих величины модуля порядка L, должно иметь место неравенство

$$\exp\left(\mathrm{Re}\left(\lambda_k\right)\tau_{\text{ПС}}\right) \le \frac{1}{N}, \quad N \gg 1,$$

т. е. они должны обладать большими по модулю отрицательными вещественными частями.

Важным моментом в формулах (6.3.12) является и то, что требования на собственные значения связаны с промежутком наблюдения решения $\left(\tau_{\text{ПС}} \ll T\right)$.

Наиболее типичен случай линейной жесткой системы, когда собственные значения матрицы отчетливо разделяются по величине их модулей на две группы. При этом собственные значения λ_k первой группы с большими модулями определяют поведение решения в пограничном слое, и соответствующие им составляющие быстро убывают, а λ_k второй группы с малыми модулями характеризуют поведение решения при $t > \tau_{\text{ПС}}$. Однако возможны и другие случаи. Например, собственные значения могут быть расположены на вещественной оси достаточно равномерно и $k(\mathbf{A}) \gg 1$. Такая система может быть жесткой, если имеет место (6.3.12).

Свойство 3. Если среди собственных значений λ_k есть кратные, то, учитывая необходимость выполнения (6.3.6) для любых начальных условий, аналогично предыдущему потребуем выполнения (6.3.6) для решения

$$x^{(l)}(t) = C_k \exp(\alpha_k t) \cdot \cos(\omega_k t + \varphi_k) P_{s-1}(t), \tag{6.3.13}$$

где $P_{s-1}(t)$ — произвольный полином ($s-1$)-ой степени, s — кратность собственного значения λ_k.

Значения C_k и φ_k выбираются при условии $\max\limits_{t \in [0,T]} \left| x^{(l)}(t) \right| = 1$. Тогда из неравенства $\left| \dfrac{dx^{(l)}}{dt} \right|_{t > \tau_{\text{ПС}}} \le \dfrac{L}{N}$ получаем требование на жорданову форму $\mathbf{J}(\mathbf{A})$ матрицы $\mathbf{A}$ жесткой системы:

$$\left\| \mathbf{J}(\mathbf{A}) \exp\left(\mathbf{J}(\mathbf{A}) t\right) \right\|_{t \ge \tau_{\text{ПС}}} \le \frac{L}{N}, \tag{6.3.14}$$

т. к. строки матрицы, стоящей в левой части неравенства (6.3.14), являются различными вариантами производных (6.3.13).

Свойство 4. В качестве еще одного свойства рассмотрим поведение фундаментальной матрицы решений $\exp(\mathbf{A}(t-t_0))$ системы (6.3.8) при изменении t влево и вправо от точки t_0.

При изменении t вправо от точки t_0 после прохождения пограничного слоя быстроизменяющиеся составляющие решения практически исчезают, и норма производной фундаментальной матрицы $\|\mathbf{A}\exp(\mathbf{A}(t-t_0))\|$ становится относительно малой. Об этом свидетельствует и неравенство (6.3.14) для ее жордановой формы.

В то же время изменение t влево от t_0 приводит к тому, что определяющую роль начинают выполнять именно быстроизменяющиеся экспоненты. Норма производной фундаментальной матрицы начинает расти экспоненциально с достаточно большим показателем. Такое поведение нормы является ярким признаком жесткости (6.3.8).

Свойство 5. Рассмотрим теперь, как связана жесткость системы (6.3.8) с жесткостью неоднородной системы (6.3.7). Решение (6.3.7) может быть записано в виде

$$\mathbf{x}(t) = \exp(\mathbf{A}(t-t_0))(\mathbf{x}(t_0) - \varphi(t_0)) + \varphi(t), \tag{6.3.15}$$

где $\varphi(t)$ — частное решение (6.3.7), определяемое видом функции $\mathbf{f}(t)$.

Пусть система (6.3.7) — жесткая. Тогда требование (6.3.6) должно выполняться при любых начальных условиях. Выбирая в частном случае $\mathbf{x}(t_0) = \varphi(t_0)$, убеждаемся, что условие (6.3.6) должно выполняться не только для самого вектора $\mathbf{x}(t)$, но и для его обоих слагаемых в (6.3.15) в отдельности. А так как первое слагаемое в (6.3.15) является решением однородной системы (6.3.8), то эта однородная система должна быть жесткой.

Таким образом, из жесткости неоднородной системы (6.3.7) следует жесткость (6.3.8). Это утверждение свидетельствует о том, что жесткость является внутренним свойством линейной системы и не может появиться только благодаря изменениям функции $\mathbf{f}(t)$.

Свойство 6. Рассмотрим поведение решения жесткой системы вне пограничного слоя. Пусть среди m собственных значений λ_k первые s значений име-

имеют большие модули, а отвечающие им частные решения быстро убывают в пределах пограничного слоя:

$$\exp\left(\mathrm{Re}(\lambda_k)\tau_{\text{ПС}}\right) \ll 1, \quad k=1,\ 2,\ \ldots,\ s. \qquad (6.3.16)$$

Записывая решение однородной системы (6.3.8) по формуле Лагранжа — Сильвестра, с учетом равенства (П3.16) получаем

$$\mathbf{x}(t) = e^{\mathbf{A}t}\mathbf{x}_0 = \sum_{j=1}^{m} e^{\lambda_j t}\mathbf{u}_j\left(\mathbf{v}_j^T\mathbf{x}_0\right), \qquad (6.3.17)$$

где $\mathbf{u}_k$ и $\mathbf{v}_k$ — собственные векторы матриц $\mathbf{A}$ и $\mathbf{A}^T$ соответственно.

Как показано в *разд. П3.3*, для этих векторов, относящихся к различным собственным значениям, имеет место равенство:

$$\mathbf{v}_k^T\mathbf{u}_j = 0, \quad \text{если } \lambda_k \neq \lambda_j.$$

Поэтому для $\mathbf{v}_k^T\mathbf{x}$ справедливо следующее выражение:

$$\mathbf{v}_k^T\mathbf{x} = \mathbf{v}_k^T e^{\mathbf{A}t}\mathbf{x}_0 = \left(\left(e^{\mathbf{A}t}\right)^T \mathbf{v}_k\right)^T \mathbf{x}_0 = e^{\lambda_k t}\cdot\mathbf{v}_k^T\mathbf{x}_0; \quad k=1,\ 2,\ \ldots,\ s.$$

Учитывая неравенство (6.3.16), вне пограничного слоя имеем почти точное равенство:

$$\mathbf{v}_k^T\mathbf{x} = 0;\ k = 1, 2, \ldots, s. \qquad (6.3.18)$$

Если среди λ_k есть кратные, то для построения аналогичных равенств используются векторы, приводящие $\mathbf{A}^T$ к жордановой форме.

Таким образом, для жесткой системы дифференциальных уравнений (6.3.8) вне пограничного слоя между компонентами вектора $\mathbf{x}(t)$ устанавливаются почти точные алгебраические связи. Их число отвечает количеству быстро убывающих частных решений s. Поэтому, выражая s-й компонент вектора $\mathbf{x}(t)$ через остальные, приходим к выводу, что вне пограничного слоя решение жесткой системы может быть описано решением системы меньшей размерности, уже не являющейся жесткой.

Вопросы практического построения коэффициентов алгебраических связей вида (6.3.18) и распространения этого свойства на нелинейные системы подробно рассматриваются в книге [40].

Многие из рассмотренных свойств решений линейной жесткой системы (6.3.8) с постоянными коэффициентами легко перенести на системы с переменной матрицей

$$\frac{d\mathbf{x}}{dt} = \mathbf{A}(t) \cdot \mathbf{x}. \tag{6.3.19}$$

Однако судить о ее жесткости по собственным значениям $\lambda_k(t)$ матрицы $\mathbf{A}(t)$ можно, если собственные векторы $\mathbf{A}(t)$ изменяются не слишком сильно. В общем случае $\lambda_k(t)$ и характеристики, определяющие рост решения (6.3.19), могут радикально отличаться. Так, например, для матрицы $\mathbf{A}(t)$ третьего порядка

$$\mathbf{A}(t) = \begin{pmatrix} -1 + 100\cos 200t & 100(1 - \sin 200t) & 0 \\ -100(1 + \sin 200t) & -(1 + 100\cos 200t) & 0 \\ 1200(\cos 100t + \sin 100t) & 1200(\cos 100t - \sin 100t) & -501 \end{pmatrix}, \ t \in [0,1]$$

собственные значения, полученные из уравнения

$$\det(\mathbf{A} - \lambda \mathbf{E}) = 0,$$

постоянны на всем отрезке решения

$$\lambda_1 = -501, \quad \lambda_2 = -1, \quad \lambda_3 = -1.$$

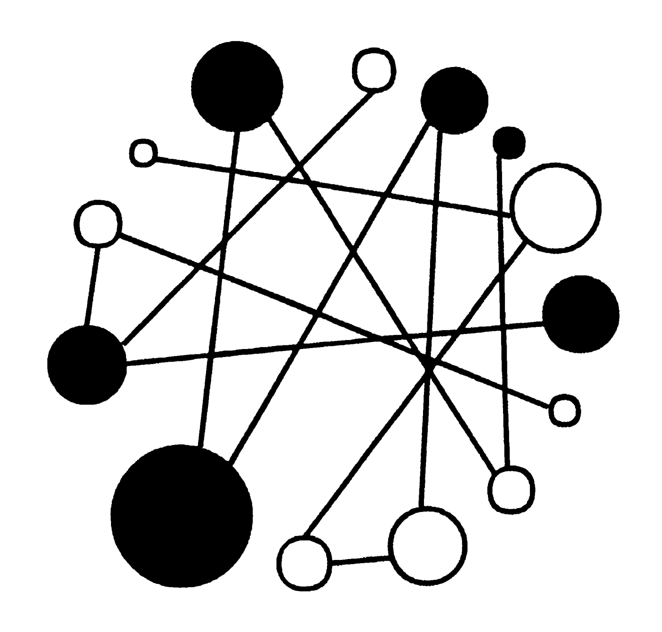
Линейную систему (6.3.8) с постоянной матрицей, обладающую такими собственными значениями, следует считать жесткой при $t \in [0, 1]$. Однако система (6.3.19) не может быть отнесена к категории жестких, что легко видеть из ее общего решения с быстрорастущими экспонентами, имеющими положительный показатель:

$$x^{(1)}(t) = C_1 e^{99t} \cos 100t + C_2 e^{-101t} \sin 100t,$$

$$x^{(2)}(t) = -C_1 e^{99t} \sin 100t + C_2 e^{-101t} \cos 100t,$$

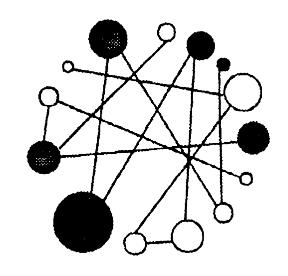
$$x^{(3)}(t) = 2C_1 e^{99t} + 3C_2 e^{-101t} + C_3 e^{-501t},$$

где постоянные C_1, C_2, C_3 определяются из начальных условий.



Приложения

ПРИЛОЖЕНИЕ 1



Конечные разности, суммы, разностные уравнения

П1.1. Конечные разности и их свойства

Пусть значения функции $f(x)$ известны лишь для дискретного множества значений независимой переменной x:

x	x_0	x_1	x_2	...	x_m
$f(x)$	$f(x_0)$	$f(x_1)$	$f(x_2)$	...	$f(x_m)$

В данном разделе значения x_k будем полагать равноотстоящими, т. е. $x_k = x_0 + kh$ (x_0, h — некоторые фиксированные числа, k — любое целое число). Величину h называют шагом таблицы. Подобно тому как поведение непрерывной и дифференцируемой функции $f(x)$ характеризуют дифференциал и производные, для таблично заданной функции вводят понятие конечной разности. Выражение

$$\Delta_h f(x_k) = f(x_k + h) - f(x_k) = f\left(x_0 + (k+1)h\right) - f\left(x_0 + kh\right) \qquad (\text{П}1.1)$$

называют *конечной разностью* (разностным оператором) первого порядка. Так как величины x_0 и h постоянны для рассматриваемой таблицы, целесообразно без нарушения общности от переменной $x_k = x_0 + kh$ перейти к новой $k = \dfrac{x_k - x_0}{h}$, которая принимает целые значения 0, 1, ..., $m-1$. Тогда

функция $f(x)$ становится функцией целочисленной переменной $f(k)$, а в операторе $\Delta_h f(x_k)$ будем опускать индекс h

$$\Delta f(k) = \Delta f_k = f(k+1) - f(k) = f_{k+1} - f_k; \qquad f_k = f(k).$$

Теперь обратимся к некоторым свойствам конечных разностей, отмечая тесную связь между ними и свойствами производных, что является основой большинства конечноразностных выражений:

- $\Delta\alpha = 0,\ \alpha = \text{const};$
- $\Delta(\alpha f(k)) = \alpha\Delta f(k);$
- $\Delta(f(k) + g(k)) = \Delta f(k) + \Delta g(k).$

Эти три свойства не требуют доказательств в силу их элементарности. Четвертое свойство необходимо пояснить

$$\begin{aligned}\Delta(f(k)\cdot g(k)) &= f(k+1)g(k+1) - f(k)g(k) = \\ &= f(k+1)g(k+1) - f(k)g(k+1) + f(k)g(k+1) - f(k)g(k) = \\ &= \Delta f(k)\cdot g(k+1) + f(k)\Delta g(k).\end{aligned}$$

Его можно записать и в таком виде

$$\Delta(f(k)\cdot g(k)) = \Delta g(k)\cdot f(k+1) + g(k)\Delta f(k).$$

Этот результат, с одной стороны, продолжает цепь аналогий со свойствами производных, т. к. формула разности произведения напоминает формулу производной произведения:

$$\left(f(x)g(x)\right)' = f'(x)g(x) + f(x)g'(x),$$

а, с другой стороны, конечность приращения аргумента подчеркивает различия.

Следующая формула имеет не столь лаконичный вид, который имеет аналогичная формула для производной, хотя первое слагаемое совпадает:

$$\Delta k^s = (k+1)^s - k^s = sk^{s-1} + \frac{s(s-1)}{2}k^{s-2} + \ldots \qquad (\text{П}1.2)$$

Однако ее результат позволяет сформулировать свойство конечной разности, созвучное свойству производной.

Конечная разность от полинома степени s равна полиному степени $s-1$.

Аналогично тому, как в непрерывном случае строилась таблица производных, рассмотрим конечные разности для наиболее популярных функций.

$$\Delta a^k = a^{k+1} - a^k = a^k(a-1),$$

$$\Delta \sin(k\alpha) = \sin((k+1)\alpha) - \sin(k\alpha) = 2\sin\left(\frac{\alpha}{2}\right)\cos\left(k\alpha + \frac{\alpha}{2}\right),$$

$$\Delta \cos\left(k\alpha - \frac{\alpha}{2}\right) = \cos\left(k\alpha + \frac{\alpha}{2}\right) - \cos\left(k\alpha - \frac{\alpha}{2}\right) = -2\sin\left(\frac{\alpha}{2}\right)\sin(k\alpha),$$

$$\Delta \frac{1}{k} = \frac{1}{k+1} - \frac{1}{k} = -\frac{1}{k(k+1)},$$

$$\Delta(-1)^k \cos\left(k\alpha - \frac{\alpha}{2}\right) = (-1)^{k+1}\cos\left(k\alpha + \frac{\alpha}{2}\right) - (-1)^k \cos\left(k\alpha - \frac{\alpha}{2}\right) =$$

$$= (-1)^{k+1}\left(\cos\left(k\alpha + \frac{\alpha}{2}\right) + \cos\left(k\alpha - \frac{\alpha}{2}\right)\right) = (-1)^{k+1} 2\cos\left(\frac{\alpha}{2}\right)\cos(k\alpha).$$

Представляет интерес первая формула из представленных для $a = 2$: $\Delta 2^k = 2^k(2-1) = 2^k$. Она показывает, что число 2 в исчислении конечных разностей в некотором отношении играет ту же роль, что и число e в дифференциальном исчислении: $(e^x)' = e^x$. Очевидно, что представленный перечень функций легко может быть продолжен. Так особую роль в теории конечных разностей играет факториальный многочлен или обобщенная разность:

$$k^{[n]} = k(k-1)(k-2)\ldots(k-n+1),$$

$$k^{[-n]} = \frac{1}{k(k+1)(k+2)\ldots(k+n-1)}, \quad k^{[0]} = 1, \quad n > 0.$$

Конечные разности в этом случае имеют вид:

$$\Delta k^{[n]} = nk^{[n-1]}, \qquad \Delta k^{[-n]} = -nk^{[-n-1]},$$

что аналогично формуле дифференцирования обычной степени в непрерывном анализе:

$$\frac{d}{dx}x^n = nx^{n-1}, \qquad \frac{d}{dx}x^{-n} = -nx^{-n-1}.$$

Подобно дифференциалам и производным высокого порядка соответствующие конечные разности строятся на основе рекуррентных соотношений. Так конечная разность порядка $s+1$ определяется следующим образом:

$$\Delta^{s+1} f_k = \Delta\left(\Delta^s f_k\right) = \Delta^s f_{k+1} - \Delta^s f_k .$$

В частности, для конечных разностей второго и третьего порядка получаем

$$\Delta^2 f_k = \Delta\left(\Delta f_k\right) = \Delta f_{k+1} - \Delta f_k = f_{k+2} - f_{k+1} - f_{k+1} + f_k = f_{k+2} - 2f_{k+1} + f_k ,$$

$$\Delta^3 f_k = \Delta\left(\Delta^2 f_k\right) = \Delta^2 f_{k+1} - \Delta^2 f_k = f_{k+3} - 3f_{k+2} + 3f_{k+1} - f_k .$$

Легко заметить в этих формулах появление коэффициентов бинома Ньютона C_s^k. По индукции можно показать, что эта тенденция в выражении конечных разностей через значения функции сохраняется:

$$\Delta^s f_k = \sum_{i=0}^{s} (-1)^i C_s^i f_{k+s-i} . \qquad (П1.3)$$

П1.2. Разделенные разности и их свойства

Для равноотстоящих узлов таблицы конечные разности являются хорошей характеристикой изменения функции, аналогичной производной для непрерывного случая. При произвольном расположении узлов таблицы:

x	x_0	x_1	x_2	...	x_m
$f(x)$	$f(x_0)$	$f(x_1)$	$f(x_2)$	...	$f(x_m)$

целесообразно ввести понятие *разделенной разности*. Разделенные разности (или разностные отношения) нулевого порядка совпадают со значениями функции, а разности первого порядка определяются равенством

$$f(x_{n-1}; x_n) = \frac{f(x_n) - f(x_{n-1})}{x_n - x_{n-1}} . \qquad (П1.4)$$

В частности, для $n=1$ и $n=2$ имеем

$$f(x_0; x_1) = \frac{f(x_1) - f(x_0)}{x_1 - x_0}, \quad f(x_1; x_2) = \frac{f(x_2) - f(x_1)}{x_2 - x_1}, \ldots$$

Аналогично строятся разделенные разности высших порядков. При этом разности k-го порядка определяются через разности ($k-1$)-го порядка по формулам

$$f(x_0;x_1;...;x_k)=\frac{f(x_1;x_2;...;x_k)-f(x_0;x_1;...;x_{k-1})}{x_k-x_0}. \qquad (\text{П}1.5)$$

В частности имеем для разностей второго порядка

$$f(x_k;x_{k+1};x_{k+2})=\frac{f(x_{k+1};x_{k+2})-f(x_k;x_{k+1})}{x_{k+2}-x_k}$$

и разностей третьего порядка

$$f(x_k;x_{k+1};x_{k+2};x_{k+3})=\frac{f(x_{k+1};x_{k+2};x_{k+3})-f(x_k;x_{k+1};x_{k+2})}{x_{k+3}-x_k}.$$

На практике вычисление разделенных разностей производится в рамках следующей таблицы, где появление новой разделенной разности более высокого порядка связано с построением еще одной диагонали:

x_0	$f(x_0)$	$f(x_0;x_1)$	$f(x_0;x_1;x_2)$	$f(x_0;x_1;x_2;x_3)$	$f(x_0;x_1;x_2;x_3;x_4)$
x_1	$f(x_1)$	$f(x_1;x_2)$	$f(x_1;x_2;x_3)$	$f(x_1;x_2;x_3;x_4)$	
x_2	$f(x_2)$	$f(x_2;x_3)$	$f(x_2;x_3;x_4)$		
x_3	$f(x_3)$	$f(x_3;x_4)$			
x_4	$f(x_4)$				

Подобно формуле (П1.3) разделенные разности могут быть выражены через значения функции в различных точках. По индукции легко убедиться в справедливости следующей формулы для разности k-го порядка

$$\begin{aligned}f(x_i;x_{i+1};...;x_{i+k})&=\frac{f(x_i)}{(x_i-x_{i+1})(x_i-x_{i+2})...(x_i-x_{i+k})}+\\&+\frac{f(x_{i+1})}{(x_{i+1}-x_i)(x_{i+1}-x_{i+2})...(x_{i+1}-x_{i+k})}+\\&+\frac{f(x_{i+k})}{(x_{i+k}-x_i)(x_{i+k}-x_{i+1})...(x_{i+k}-x_{i+k-1})}+...\end{aligned} \qquad (\text{П}1.6)$$

$$f(x_i;x_{i+1};...;x_{i+k})=\sum_{j=i}^{i+k}\frac{f(x_j)}{\prod\limits_{l\neq j}(x_l-x_j)}.$$

Непосредственно из этой формулы следует важное свойство разделенных разностей: они являются симметричными функциями своих аргументов.

В частности, $f(x_n;x_{n-1}) = f(x_{n-1};x_n)$.

Если в исходной таблице узлы равноотстоящие, то для описания поведения таблично заданной функции могут быть привлечены как разделенные разности, так и конечные. Установим связь между ними для этого случая.

$$f(x_i;x_{i+1}) = \frac{f(x_{i+1}) - f(x_i)}{x_{i+1} - x_i} = \frac{\Delta f_i}{h},$$

$$f(x_i;x_{i+1};x_{i+2}) = \frac{\dfrac{\Delta f_{i+1}}{h} - \dfrac{\Delta f_i}{h}}{2h} = \frac{\Delta^2 f_i}{2h^2}.$$

По индукции легко устанавливается, что

$$f(x_i;x_{i+1};...;x_{i+k-1};x_{i+k}) = \frac{\Delta^k f_i}{k!h^k}. \qquad (\text{П}1.7)$$

П1.3. Суммирование функций

Обратимся к уравнению

$$\Delta F(k) = \varphi(k). \qquad (\text{П}1.8)$$

До сих пор мы занимались прямой задачей: по заданной функции $F(k)$ необходимо определить функцию $\varphi(k)$. Теперь обратимся к обратной задаче: по заданной функции $\varphi(k)$ необходимо восстановить функцию $F(k)$. Ситуация подобна задаче нахождения функции $f(x)$ по ее производной $h(x)$.

$$f'(x) = h(x).$$

В случае ее решения появляется возможность для вычисления интеграла

$$\int_a^b h(x)dx = \int_a^b f'(x)dx = f(b) - f(a). \qquad (\text{П}1.9)$$

Аналогично, решение обратной задачи (П1.8) позволяет, в свою очередь, успешно решать задачу суммирования функции $\varphi(k)$.

Запишем уравнение (П1.8) последовательно для $k = m, m+1, ..., N-1$ и результаты просуммируем:

$$F(m+1) - F(m) = \varphi(m),$$

$$F(m+2) - F(m+1) = \varphi(m+1),$$

$$F(m+3) - F(m+2) = \varphi(m+2),$$

$$\ldots$$

$$F(N) - F(N-1) = \varphi(N-1);$$

$$F(N) - F(m) = \sum_{k=m}^{N-1} \varphi(k).$$

Иными словами

$$\sum_{k=m}^{N-1} \varphi(k) = \sum_{k=m}^{N-1} \Delta F(k) = F(N) - F(m). \qquad (\text{П}1.10)$$

Эта формула является дискретным аналогом формулы Ньютона — Лейбница (П1.9). В дополнение следует заметить, что она выводилась в предположении, что $N > m$.

Простейшие формулы интегрирования получаются интегрированием таблицы производных. Например,

$$\left(x^{m+1}\right)' = (m+1)x^m \quad \Rightarrow \quad \int\left(x^{m+1}\right)' dx = (m+1)\int x^m dx \quad \Rightarrow \quad \int x^m dx = \frac{x^{m+1}}{m+1}.$$

Аналогично, обращаясь к таблице конечных разностей, получаем

$$\Delta a^k = a^k(a-1) \quad \Rightarrow \quad \sum_{k=m}^{N-1} \Delta a^k = \sum_{k=m}^{N-1} a^k(a-1) \quad \Rightarrow \quad \sum_{k=m}^{N-1} a^k = \frac{\left(a^N - a^m\right)}{a-1}.$$

Второй пример:

$$\Delta \sin(k\alpha) = \sin\left((k+1)\alpha\right) - \sin(k\alpha) = 2\sin\left(\frac{\alpha}{2}\right)\cos\left(k\alpha + \frac{\alpha}{2}\right),$$

$$\cos\left(k\alpha + \frac{\alpha}{2}\right) = \frac{\Delta \sin(k\alpha)}{2\sin\left(\frac{\alpha}{2}\right)} \quad \Rightarrow \quad \sum_{k=p}^{n-1} \cos\left(k\alpha + \frac{\alpha}{2}\right) = \frac{\sin(n\alpha) - \sin(p\alpha)}{2\sin\left(\frac{\alpha}{2}\right)}.$$

Обращая и другие формулы разностей, можно построить таблицу сумм, напоминающую таблицу простейших интегралов.

Весьма популярным приемом в интегральном исчислении является также интегрирование по частям. Его аналогом в дискретном случае является суммирование по частям.

Обратимся сначала к интегралам. Введем три функции — $u(t)$, $v(t)$ и $U(t)$:

$$U(t)=\int_0^t u(t)dt,$$

а затем продифференцируем произведение $U(t)v(t)$:

$$\frac{d}{dt}\big(U(t)v(t)\big)=\frac{dU(t)}{dt}v(t)+U(t)\frac{dv(t)}{dt}=u(t)v(t)+U(t)\frac{dv(t)}{dt}.$$

Перегруппируем члены равенства и проинтегрируем от a до b:

$$\begin{aligned}\int_a^b u(t)v(t)dt &= \int_a^b \frac{d}{dt}\big(U(t)v(t)\big)dt-\int_a^b U(t)\frac{dv(t)}{dt}dt = \\ &= U(t)v(t)\Big|_{t=a}^{t=b}-\int_a^b U(t)\frac{dv(t)}{dt}dt.\end{aligned} \qquad (\text{П}1.11)$$

Теперь обратимся к суммированию по частям. Аналогично введем три функции — $u(k)$, $v(k)$ и $U(k)$:

$$U(k)=\sum_{i=0}^{k}u(i).$$

Учитывая, что

$$\Delta U(k)=U(k+1)-U(k)=u(k+1),$$

$$\sum_{k=p}^{N}\Delta\big(U(k)v(k)\big)=U(N+1)v(N+1)-U(p)v(p),$$

воспользуемся ранее полученной формулой конечной разности для произведения

$$\Delta\big(U(k)v(k)\big)=v(k+1)\Delta U(k)+U(k)\Delta v(k)=v(k+1)u(k+1)+U(k)\Delta v(k).$$

Перегруппируем члены этого равенства и просуммируем обе его части:

$$\sum_{k=p}^{N}u(k+1)v(k+1)=U(k)v(k)\Big|_{k=p}^{k=N+1}-\sum_{k=p}^{N}U(k)\Delta v(k). \qquad (\text{П}1.12)$$

Это и есть формула суммирования по частям или *формула Абеля*. Можно исключить из нее $U(k)$, выразив эту функцию через $u(k)$:

$$\sum_{k=p}^{N} u(k+1)v(k+1) = \left(\sum_{i=0}^{k} u(i)\right)v(k)\Bigg|_{k=p}^{k=N+1} - \sum_{k=p}^{N}\left(\sum_{i=0}^{k} u(i)\right)\Delta v(k).$$

Представляется весьма полезным сравнение формул (П1.11) и (П1.12).

Обратимся к самому популярному примеру суммирования на основе формулы Абеля. Требуется найти сумму

$$S = \sum_{k=1}^{N} ka^k.$$

Вводя обозначения $u(k) = a^{k-1}$ и $v(k) = k-1$ и вычисляя $U(k)$:

$$U(k) = \sum_{i=0}^{k} u(i) = \sum_{i=0}^{k} a^{i-1} = \frac{1}{a}\frac{a^{k+1}-1}{a-1},$$

с учетом $\Delta v(k) = 1$ непосредственно по (П1.12) получаем

$$S = \sum_{k=1}^{N} ka^k = \frac{1}{a}\frac{a^{N+2}-1}{a-1}N - 0 - \sum_{k=1}^{N}\frac{1}{a}\frac{a^{k+1}-1}{a-1} =$$

$$= \frac{1}{a}\frac{a^{N+2}-1}{a-1}N - 0 + \frac{N}{a(a-1)} - \frac{1}{(a-1)}\sum_{k=1}^{N} a^k = \frac{a^{N+1}}{a-1}\cdot N - \frac{a^{N+1}-a}{(a-1)^2}.$$

П1.4. Разностные уравнения

Первоначально обратимся к дифференциальным уравнениям. Соотношение

$$F\left(t, z(t), z'(t), \ldots, z^{(s)}(t)\right) = 0,$$

где t — независимая переменная, функция F задана, функция $z(t)$ — искомая, называется *дифференциальным уравнением порядка* s. При этом уравнение может быть разрешено относительно старшей производной

$$z^{(s)}(t) = f\left(t, z(t), z'(t), \ldots, z^{(s-1)}(t)\right). \qquad \text{(П1.13)}$$

Порядок уравнения s, определяемый номером старшей производной, является важной характеристикой уравнения (П1.13). Так он определяет количество

начальных условий, необходимых для однозначного решения (П1.13). Если дифференциальное уравнение является линейным относительно функции $z(t)$ и ее производных, то величина s задает количество линейно независимых решений и т. д.

Рассмотрим разностный аналог дифференциального уравнения

$$F\left(k, f(k), \Delta f(k), \Delta^2 f(k), ..., \Delta^s f(k)\right) = 0, \qquad (П1.14)$$

где k — независимая целочисленная переменная, функция F задана, функция $f(k)$ — искомая.

Казалось бы, логично считать порядок этого уравнения равным s, руководствуясь номером старшей конечной разности, как это было с производными в уравнении (П1.13). Рассмотрим, однако, следующий пример:

$$2\Delta^3 f_k + 3\Delta^2 f_k - f_k = 0, \qquad f_k \equiv f(k).$$

Выразим все конечные разности через значения функции в различных точках, руководствуясь уравнением (П1.3):

$$\begin{gathered} 2\left(f_{k+3} - 3f_{k+2} + 3f_{k+1} - f_k\right) + 3\left(f_{k+2} - 2f_{k+1} + f_k\right) - f_k = 0, \\ 2f_{k+3} - 3f_{k+2} = 0. \end{gathered} \qquad (П1.15)$$

Задаваясь только *одним* начальным условием f_0 (вместо ожидаемых трех) и последовательно полагая в (П1.15) значение $k = -2, -1, 0, 1, ...$, шаг за шагом воспроизводим f_k для любого значения k. Здесь лежит глубокое различие между дифференциальными и разностными уравнениями. Легко заметить, что эффект снижения ожидаемого порядка уравнения произошел за счет сокращения слагаемых f_k и f_{k+1} в (П1.15). По этой причине в общем случае для определения порядка разностного уравнения будем выражать все конечные разности через значения функции. Тогда после всех упрощений порядок разностного уравнения будет определяться разностью между наибольшим и наименьшим значениями аргумента функции $f(k)$. В дальнейшем будем записывать разностные уравнения в виде

$$\Phi\left(k, f(k), f(k+1), ..., f(k+s)\right) = 0$$

или в виде, разрешенном относительно функции с наибольшим значением аргумента

$$f(k+s) = \Phi_1\left(k, f(k), ..., f(k+s-1)\right). \qquad (П1.16)$$

Видно, что оба эти уравнения имеют порядок $s=(k+s)-s$. Для их однозначного решения необходимо задать s начальных условий $f(0)$, $f(1)$, $f(s-2)$, $f(s-1)$.

Перейдем к решению уравнения (П1.16). Для дифференциального уравнения (П1.13) традиционно большое внимание уделяется теоремам существования и единственности решения. Для весьма узкого класса этих уравнений удается получить аналитическое решение. Иное положение дел имеет место для (П1.16). Последовательно полагаем здесь $k=0, 1, 2, \dots$

$$f(s)=\Phi_1\left(0, f(0), \dots, f(s-1)\right),$$

$$f(s+1)=\Phi_1\left(1, f(1), \dots, f(s)\right),$$

$$f(s+2)=\Phi_1\left(2, f(2), \dots, f(s+1)\right),$$

$$\dots$$

и получаем решение для всех целых значений аргумента. Такое построение решения называют *пошаговым методом решения разностного уравнения*, который всегда дает решение, когда заданы s начальных условий. Пошаговый метод совместно с начальными условиями обеспечивает существование и единственность решения начальной задачи для разностного уравнения (аналог задачи Коши для дифференциального уравнения). Важно заметить, что этот метод легко и эффективно реализуется при программировании на компьютере.

Наличие такого универсального подхода к решению (П1.16) не снижает актуальности постановки задачи аналитического решения (П1.16), хотя бы для некоторых уравнений частного вида. Это, в частности, позволило бы при необходимости вычислять непосредственно, например, $f(2007)$, минуя обязательное вычисление $f(1)$, ..., $f(2006)$ при пошаговом методе. К числу таких уравнений, допускающих построение аналитического решения, относятся линейные уравнения, в частности, с постоянными коэффициентами.

П1.4.1. Линейное разностное уравнение первого порядка

Обратимся к уравнению

$$y(k+1)=b(k)y(k)+g(k), \qquad y(0)=y_0, \qquad (\text{П}1.17)$$

где $g(k)$ и $b(k)$ — заданные функции k, $y(k)$ — искомая функция. Если $g(k) \equiv 0$, уравнение называется *однородным*, в противном случае — *неоднородным*. Однородное уравнение имеет вид:

$$u(k+1) = b(k)u(k), \qquad u(0) = u_0.$$

Воспользуемся пошаговым методом:

$$u(1) = b(0)u_0,$$

$$u(2) = b(1)u(1) = b(0)b(1)u_0,$$

$$u(3) = b(2)u(2) = b(0)b(1)b(2)u_0,$$

$$\ldots$$

$$u(n) = b(n-1)u(n-1) = b(0)b(1)\ldots b(n-1)u_0 = \left(\prod_{k=0}^{n-1} b(k)\right)u_0.$$

Если, в частности, $b(k) = \text{const} = b$, то решение имеет вид $u(n) = b^n u_0$.

Решение неоднородного уравнения будем искать в виде

$$y(k) = u(k)v(k),$$

где $u(k)$ — решение однородного уравнения, а $v(k)$ — произвольная функция. Подставим искомый вид решения в уравнение (П1.17):

$$u(k+1)v(k+1) = b(k)u(k)v(k) + g(k).$$

Выполним очевидные преобразования:

$$u(k+1)v(k+1) - u(k+1)v(k) + u(k+1)v(k) - b(k)u(k)v(k) = g(k),$$

$$u(k+1)\Delta v(k) + \left[u(k+1) - b(k)u(k)\right]v(k) = g(k).$$

В квадратных скобках стоит однородное уравнение, в которое подставлено его решение, что обращает скобку в ноль, и тогда

$$\Delta v(k) = u^{-1}(k+1)g(k).$$

Просуммируем это равенство от 0 до $k-1$:

$$v(k) = v(0) + \sum_{i=0}^{k-1} u^{-1}(i+1)g(i).$$

Подставим полученное в $y(k)$, учитывая, что $u(k)=\left(\prod_{\chi=0}^{k-1} b(\chi)\right)u_0$:

$$y(k)=u(k)v(k)=$$
$$=\left(\prod_{\chi=0}^{k-1} b(\chi)\right)u_0 v(0)+\left(\prod_{\chi=0}^{k-1} b(\chi)\right)u_0\sum_{\chi=0}^{k-1}\left(\prod_{\lambda=0}^{\chi} b(\lambda)\right)^{-1} u_0^{-1} g(\chi)= \tag{П1.18}$$
$$=\left(\prod_{\chi=0}^{k-1} b(\chi)\right)\left(y(0)+\sum_{\chi=0}^{k-1}\left(\prod_{\lambda=0}^{\chi} b(\lambda)\right)^{-1} g(\chi)\right).$$

Для уравнения с постоянными коэффициентами, когда $b(k)=\text{const}=b$, решение имеет вид:

$$y(k)=b^k\left(y_0+\sum_{\chi=0}^{k-1} b^{-\chi-1}g(\chi)\right)=b^k y_0+\sum_{\chi=0}^{k-1} b^{k-\chi-1}g(\chi)=b^k y_0+\sum_{\chi=0}^{k-1} b^{\chi}g(k-\chi-1).$$

Если дополнительно $g(k)=\text{const}=g$, то

$$y(k)=b^k y(0)+\left(\sum_{\chi=0}^{k-1} b^{\chi}\right)g=b^k y(0)+\frac{1-b^k}{1-b}g. \tag{П1.19}$$

Как известно, решение линейного дифференциального уравнения

$$\frac{dz(t)}{dt}=a(t)z(t)+g(t), \qquad z(t_0)=z_0$$

имеет вид

$$z(t)=e^{\int_0^t a(\tau)d\tau}\left(z_0+\int_{t_0}^{t} e^{-\int_0^{\tau} a(\xi)d\xi} g(\tau)d\tau\right).$$

Переписав решение линейного разностного уравнения

$$y(k)=e^{\ln\prod_{\chi=0}^{k-1} b(\chi)}\left(y(0)+\sum_{\chi=0}^{k-1} e^{-\ln\prod_{\lambda=0}^{\chi-1} b(\lambda)} g(\chi)\right),$$

получим еще одно звено в цепочке аналогий двух исчислений.

П1.4.2. Линейные разностные уравнения порядка выше первого

Перейдем к уравнению s-го порядка:

$$y(k+s)+b_1(k)y(k+s-1)+b_2(k)y(k+s-2)+...+b_s(k)y(k)=g(k), \quad (\text{П}1.20)$$

где $g(k)$ и $b_i(k)$ — заданные функции k, $y(k)$ — искомая функция.

Ряд теорем, аналогичных теоремам для дифференциальных уравнений, устанавливают свойства частных и общих решений линейных разностных уравнений. Ограничимся лишь формулировкой этих теорем.

Теорема 1. Если $y_1(k)$, $y_2(k)$, $y_p(k)$ — частные решения линейного однородного уравнения

$$y(k+s)+b_1(k)y(k+s-1)+b_2(k)y(k+s-2)+...+b_s(k)y(k)=0, \quad (\text{П}1.21)$$

то и функция

$$x(k)=c_1y_1(k)+c_2y_2(k)+...+c_py_p(k),$$

где c_1, c_2, ..., c_p — произвольные постоянные, также будет частным решением этого уравнения.

Теорема 2. Если s частных решений однородного уравнения $y_1(k)$, $y_2(k)$, $y_s(k)$ линейно независимы, то

$$y(k)=\sum_{i=1}^{s}c_iy_i(k) \quad (\text{П}1.22)$$

является общим решением однородного уравнения.

Теорема 3. Общее решение линейного неоднородного уравнения (П1.20) представляется в виде суммы частного его решения $y_{\text{частн}}(k)$ и общего решения линейного однородного уравнения (П1.22):

$$y(k)=y_{\text{частн}}(k)+c_1y_1(k)+c_2y_2(k)+...+c_py_p(k).$$

В дальнейшем ограничимся лишь уравнениями с постоянными коэффициентами, когда в (П1.20) $b_i(k)=\text{const}=b_i$:

$$y(k+s)+b_1y(k+s-1)+b_2y(k+s-2)+...+b_sy(k)=g(k). \quad (\text{П}1.23)$$

Рассмотрим однородное уравнение

$$u(k+s)+b_1u(k+s-1)+b_2u(k+s-2)+\ldots+b_su(k)=0. \qquad (\text{П}1.24)$$

Частные решения будем искать в виде $u(k)=C\gamma^k$, где $C=\text{const}$. Такой вид искомого решения подсказывает решение уравнения первого порядка с постоянным коэффициентом $u(n)=b^n u_0$. Здесь уместно напомнить, что в дифференциальном уравнении s-ого порядка с постоянными коэффициентами

$$z^{(s)}(t)+a_1z^{(s-1)}(t)+\ldots+a_sz(t)=0$$

частные решения ищутся в форме $z(t)=\exp(\lambda_k t)$. Подставим $u(k)=C\gamma^k$ в уравнение (П1.24) и после сокращения на γ^k (мы изучаем лишь нетривиальные решения, и поэтому $\gamma\neq 0$) получаем характеристическое уравнение

$$\gamma^s+b_1\gamma^{s-1}+\ldots+b_s=0. \qquad (\text{П}1.25)$$

Это уравнение имеет s корней (с учетом их кратности): γ_1, γ_2, ..., γ_s. В зависимости от алгебраической природы корней характеристического уравнения (вещественные, комплексные, кратные) разными будут и частные решения, соответствующие каждому корню γ_r.

1. Каждому простому вещественному корню γ_r соответствует частное решение $u_r(k)=c_r\gamma_r^k$, являющееся одним из слагаемых в общем решении.
2. Каждой простой паре комплексно-сопряженных корней $\gamma_{r,r+1}=(\alpha_r\pm i\beta_r)$ соответствуют комплексные частные решения, являющиеся линейно независимыми:

   $$u_r(k)=(\alpha_r+i\beta_r)^k, \qquad u_{r+1}(k)=(\alpha_r-i\beta_r)^k$$

   или вещественные частные решения

   $$u_r(k)=\rho_r^k\cos(k\varphi_r), \quad u_{r+1}(k)=\rho_r^k\sin(k\varphi_r), \quad \rho_r=\sqrt{\alpha_r^2+\beta_r^2}, \quad \operatorname{tg}(\varphi_r)=\frac{\beta_r}{\alpha_r}.$$

   В общем решении им сопоставляются два слагаемых (вещественный вариант):

   $$c_r\rho_r^k\cos(k\varphi_r)+c_{r+1}\rho_r^k\sin(k\varphi_r).$$

3. Если среди корней встречаются кратные, то корню γ_r кратности p соответствуют частные решения:

$$u_r(k)=\gamma_r^k,\quad u_{r+1}(k)=k\gamma_r^k,\ \ldots,u_{r+p-1}(k)=k^{p-1}\gamma_r^k.$$

Решения эти линейно независимы, и в общем решении им сопоставляются слагаемые

$$c_r\gamma_r^k+c_{r+1}k\gamma_r^k+\ldots+c_{r+p-1}k^{p-1}\gamma_r^k=Q_{p-1}(k)\gamma_r^k,$$

где $Q_{p-1}(k)$ — полином от k степени $p-1$.

Общее решение однородного уравнения в соответствии с теоремой 2 является линейной комбинацией частных решений, $u_r(k)$:

$$u(k)=\sum_{r=1}^{s}c_r u_r(k).$$

Обратимся к неоднородному уравнению (П1.23). Если его правая часть $g(k)$ представляет собой линейную комбинацию полиномов от k, показательных функций, синусов или косинусов, то частное решение, как и в случае дифференциального уравнения, можно подобрать в такой же форме. Универсальным подходом по-прежнему остается метод вариации произвольных постоянных Лагранжа. Суть его состоит в поиске решения в таком же виде, как и решение однородного уравнения, но при условии, что c_r являются функциями независимой переменной $c_r=c_r(k)$.

$$y(k)=\sum_{r=1}^{s}c_r(k)u_r(k),$$

где $u_r(k)$ — частные решения однородного уравнения, а $c_r(k)$ — искомые функции. Для нахождения этих функций надо сформулировать s условий. Само уравнение дает лишь одно условие, остальные $s-1$ условия могут быть выбраны произвольно. Будем требовать, чтобы решение $y(k+l)$ и в произвольной точке $k+l$ было по структуре такое же, как и в точке k (линейная комбинация частных решений):

$$y(k+l)=\sum_{r=1}^{S}c_r(k)y_r(k+l).$$

Итак, вычисляем решение в точке $k+1$:

$$\begin{aligned} y(k+1) &= c_1(k+1)y_1(k+1) + c_2(k+1)y_2(k+1) + \ldots + c_s(k+1)y_s(k+1) - \\ &- c_1(k)y_1(k+1) - c_2(k)y_2(k+1) - \ldots - c_s(k)y_s(k+1) + \\ &+ c_1(k)y_1(k+1) + c_2(k)y_2(k+1) + \ldots + c_s(k)y_s(k+1) = \\ &= \Delta c_1(k)y_1(k+1) + \Delta c_2(k)y_2(k+1) + \ldots + \Delta c_s(k)y_s(k+1) + \\ &+ c_1(k)y_1(k+1) + c_2(k)y_2(k+1) + \ldots + c_s(k)y_s(k+1). \end{aligned}$$

Для выполнения требования о совпадении структур решений надо положить

$$(1) \qquad \Delta c_1(k)y_1(k+1) + \Delta c_2(k)y_2(k+1) + \ldots + \Delta c_s(k)y_s(k+1) = 0 .$$

Это и есть первое условие, наложенное на $y(k)$, и тогда

$$y(k+1) = c_1(k)y_1(k+1) + c_2(k)y_2(k+1) + \ldots + c_s(k)y_s(k+1) .$$

Вычисляем решение в точке $k+2$:

$$\begin{aligned} y(k+2) &= c_1(k+1)y_1(k+2) + c_2(k+1)y_2(k+2) + \ldots + c_s(k+1)y_s(k+2) - \\ &- c_1(k)y_1(k+2) - c_2(k)y_2(k+2) - \ldots - c_s(k)y_s(k+2) + \\ &+ c_1(k)y_1(k+2) + c_2(k)y_2(k+2) + \ldots + c_s(k)y_s(k+2) = \\ &= \Delta c_1(k)y_1(k+2) + \Delta c_2(k)y_2(k+2) + \ldots + \Delta c_s(k)y_s(k+2) + \\ &+ c_1(k)y_1(k+2) + c_2(k)y_2(k+2) + \ldots + c_s(k)y_s(k+2). \end{aligned}$$

Снова выполняем требование об одинаковости структур и получаем второе условие

$$(2) \qquad \Delta c_1(k)y_1(k+2) + \Delta c_2(k)y_2(k+2) + \ldots + \Delta c_s(k)y_s(k+2) = 0 ,$$

Продолжая вычисление решений в точках $k+3$, $k+4$ и т. д. до $k+s-1$, аналогичным образом получаем остальные условия:

$$(3) \qquad \Delta c_1(k)y_1(k+3) + \Delta c_2(k)y_2(k+3) + \ldots + \Delta c_s(k)y_s(k+3) = 0,$$

$$\ldots$$

$$(s-1) \quad \Delta c_1(k)y_1(k+s-1) + \Delta c_2(k)y_2(k+s-1) + \ldots + \Delta c_s(k)y_s(k+s-1) = 0.$$

И, наконец, последнее условие получаем из самого уравнения.

Вычислим решение в точке $k+s$:

$$\begin{aligned} y(k+s) &= \Delta c_1(k)y_1(k+s) + \Delta c_2(k)y_2(k+s) + \ldots + \Delta c_s(k)y_s(k+s) + \\ &+ c_1(k)y_1(k+s) + c_2(k)y_2(k+s) + \ldots + c_s(k)y_s(k+s) \end{aligned}$$

и подставим его и решения во всех предыдущих точках в уравнение (П1.23):

$$\Delta c_1(k)y_1(k+s)+\Delta c_2(k)y_2(k+s)+...+\Delta c_s(k)y_s(k+s)+$$
$$+c_1(k)y_1(k+s)+c_2(k)y_2(k+s)+...+c_s(k)y_s(k+s)+$$
$$+b_1c_1(k)y_1(k+s-1)+b_1c_2(k)y_2(k+s-1)+...+b_1c_s(k)y_s(k+s-1)+$$
$$+b_2c_1(k)y_1(k+s-2)+b_2c_2(k)y_2(k+s-2)+...+b_2c_s(k)y_s(k+s-2)+$$
$$...$$
$$+b_sc_1(k)y_1(k)+b_sc_2(k)y_2(k)+...+b_sc_s(k)y_s(k)=g(k).$$

Если в каждом r-ом столбце (без учета первой строки) вынести за скобку $c_r(k)$, то в скобках остается левая часть однородного уравнения

$$y_r(k+s)+b_1y_r(k+s-1)+b_2y_r(k+s-2)+...+b_sy_r(k),$$

в которое подставлено его частное решение, т. е. все скобки обращаются в ноль, и остается лишь

$$(s)\quad \Delta c_1(k)y_1(k+s)+\Delta c_2(k)y_2(k+s)+...+\Delta c_s(k)y_s(k+s)=g(k),$$

что является s-м условием.

Все s условий в совокупности

$$\Delta c_1(k)y_1(k+1)+\Delta c_2(k)y_2(k+1)+...+\Delta c_s(k)y_s(k+1)=0$$
$$\Delta c_1(k)y_1(k+2)+\Delta c_2(k)y_2(k+2)+...+\Delta c_s(k)y_s(k+2)=0$$
$$...$$
$$\Delta c_1(k)y_1(k+s)+\Delta c_2(k)y_2(k+s)+...+\Delta c_s(k)y_s(k+s)=g(k)$$

образуют линейную алгебраическую систему относительно искомых функций $\Delta c_r(k)$. Определитель этой системы отличен от нуля, т. к. образован из линейно независимых частных решений однородного уравнения, и, значит, система имеет единственное решение. Решение этой системы можно представить в виде:

$$\Delta c_r(k)=R_r(k)g(k),$$

где $R_r(k)$ — некоторая функция, вычисленная при решении линейной системы, а $g(k)$ — правая часть исходного уравнения.

Просуммируем последнее равенство:

$$c_r(k) = \sum_{i=0}^{k-1} R_r(i)g(i) + c_r(0)$$

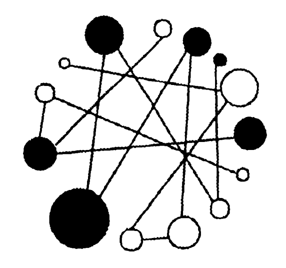
и построим общее решение неоднородного разностного уравнения:

$$y_{\text{о.н.}}(k) = \sum_{r=1}^{n}\left(\sum_{i=0}^{k-1} R_r(i)g(i) + c_r(0)\right)u_r(k).$$

Величины $c_r(0)$ являются произвольными постоянными. Используя начальные условия, можно найти их и получить частное решение.

В заключение отметим, что более полно с материалом данного раздела можно ознакомиться в книге [23].

Приложение 2



Линейные (векторные) пространства

Идея линейности является одним из важнейших принципов математики. На этой основе построены классический анализ и вариационное исчисление. Более того, многие физические процессы в малом (в определенном смысле) являются линейными, что позволяет делать о них достаточно точные выводы, изучая линейный, гораздо более простой для исследования процесс или объект.

Определение. *Линейным пространством* называется множество $\mathbf{R}$ элементов произвольной природы, именуемых векторами, удовлетворяющее следующим аксиомам.

1. Каждой паре $\mathbf{x}$ и $\mathbf{y}$ векторов из $\mathbf{R}$ ставится в соответствие вектор $\mathbf{x}+\mathbf{y}$, называемый суммой $\mathbf{x}$ и $\mathbf{y}$, причем:
   - $\mathbf{x}+\mathbf{y}=\mathbf{y}+\mathbf{x}$ (коммутативность сложения);
   - $\mathbf{x}+(\mathbf{y}+\mathbf{z})=(\mathbf{x}+\mathbf{y})+\mathbf{z}$ (ассоциативность сложения);
   - в $\mathbf{R}$ существует однозначно определенный вектор $\mathbf{0}$ (называемый *нулевым*) такой, что $\mathbf{x}+\mathbf{0}=\mathbf{x}$ для каждого $\mathbf{x}$ из $\mathbf{R}$;
   - каждому вектору $\mathbf{x}\in\mathbf{R}$ отвечает однозначно определенный вектор $-\mathbf{x}$ (называемый *противоположным*) такой, что $\mathbf{x}+(-\mathbf{x})=\mathbf{0}$.
2. Каждой паре α и $\mathbf{x}$, где α — скаляр из некоторого поля $\mathbf{K}$, $\mathbf{x}$ — вектор из $\mathbf{R}$, ставится в соответствие вектор $\alpha\mathbf{x}$, называемый *произведением* α и $\mathbf{x}$, причем:
   - $\alpha(\beta\mathbf{x})=(\alpha\beta)\mathbf{x}$ (ассоциативность умножения вектора на скаляры);
   - $1\mathbf{x}=\mathbf{x}$ для каждого $\mathbf{x}$.

3. Справедливо:

- $\alpha(\mathbf{x}+\mathbf{y})=\alpha\mathbf{x}+\alpha\mathbf{y}$ (дистрибутивность умножения на скаляры относительно сложения векторов);
- $(\alpha+\beta)\mathbf{x}=\alpha\mathbf{x}+\beta\mathbf{x}$ (дистрибутивность умножения на векторы относительно сложения векторов).

Эти аксиомы не претендуют на логическую независимость, но являются удобной характеристикой объектов, с которыми часто приходится иметь дело.

Введенные операции называют *внутренней* (сложение векторов) и *внешней* (умножение вектора на скаляр). Что представляет собой поле $\mathbf{K}$, из которого берутся скаляры для образования линейного пространства? Это наиболее известная из алгебраических структур.

Определение. *Полем скаляров* $\mathbf{K}$ называется множество объектов, в котором введены две внутренние операции: сложение (аддитивная операция) и умножение (мультипликативная операция), удовлетворяющие аксиомам, которые перечислены в табл. П2.1.

Таблица П2.1. *Аксиомы сложения и умножения*

Сложение	**Умножение**
Коммутативность	
1. $\alpha+\beta=\beta+\alpha$	1. $\alpha\beta=\beta\alpha$
Ассоциативность	
2. $\alpha+(\beta+\gamma)=(\alpha+\beta)+\gamma$	2. $(\alpha\beta)\gamma=\alpha(\beta\gamma)$
Существование нейтрального элемента	
3. $\alpha+0=\alpha$	3. $\alpha\cdot 1=\alpha$
Существование противоположного (обратного) элемента	
4. $\alpha+(-\alpha)=0$	4. $\alpha\cdot\alpha^{-1}=1$
Дистрибутивность умножения относительно сложения	
5. $\alpha(\beta+\gamma)=\alpha\beta+\alpha\gamma$	

Про векторные пространства говорят, что они образуются над некоторым полем по природе элементов этого поля: векторное пространство над полем рациональных (вещественных, комплексных) чисел или векторное пространство над комплексным (рациональным, вещественным) полем.

Разнообразные линейные пространства получаются, если конкретизировать природу образующих пространство объектов и операций с ними. Приведем примеры некоторых линейных пространств, с которыми в той или иной мере приходится сталкиваться.

1. Простейшим примером служат направленные отрезки на прямой, на плоскости и в обычном пространстве. В физике с помощью таких отрезков, называемых векторами, изображают силы, скорости, ускорения и т. д.

   - На прямой сумма — это векторы, расположенные друг за другом, а произведение вектора на скаляр — это вектор, измененной в $|\alpha|$ раз длины и с тем же (при $\alpha > 0$) или противоположным (при $\alpha < 0$) направлением. Все аксиомы очевидно выполняются.
   - На плоскости сумма двух векторов — это вектор, представляющий собой диагональ параллелограмма, построенного на векторах-слагаемых. Относительно умножения вектора на скаляр можно повторить то же самое, что сказано в предыдущем пункте.
   - В обыкновенном физическом пространстве сумма — диагональ параллелепипеда, а произведение на скаляр, как в предыдущих двух пунктах.

2. Введение декартовой системы координат позволяет устанавливать взаимно однозначное соответствие между физическими векторами, как направленными отрезками, и упорядоченными наборами скаляров, состоящими их одного, двух и трех скаляров (для векторов на прямой, плоскости и в обыкновенном пространстве соответственно). Нетрудно сделать следующий шаг и любую упорядоченную последовательность скаляров $\mathbf{x} = (x_1, x_2, x_3, ..., x_n)$ представить вектором, определив внутреннюю и внешнюю операции следующим образом:

   $$\mathbf{x} + \mathbf{y} = (x_1 + y_1, x_2 + y_2, x_3 + y_3, ..., x_n + y_n),\ \alpha\mathbf{x} = (\alpha x_1, \alpha x_2, \alpha x_2, ..., \alpha x_n).$$

   Выполнение аксиом линейного пространства легко проверяется.

   Следует отметить, что именно этот пример является основным в теории линейных пространств. Линейные пространства такого типа часто называют *арифметическими линейными пространствами*.

3. Линейное пространство образуют алгебраические полиномы степеней, не превосходящих натурального числа n:

$$p(t) = a_1 + a_2 t + a_3 t^2 + \ldots + a_{n+1} t^n, \quad q(t) = b_1 + b_2 t + b_3 t^2 + \ldots + b_{n+1} t^n.$$

   Операции с полиномами ничем не отличаются от известных из школьной алгебры:

$$p(t) + q(t) = (a_1 + b_1) + (a_2 + b_2)t + (a_3 + b_3)t^2 + \ldots + (a_{n+1} + b_{n+1})t^n,$$

$$\alpha p(t) = \alpha a_1 + \alpha a_2 t + a_3 t^2 + \ldots + \alpha a_{n+1} t^n.$$

   Заметим, что полиномы конкретной степени n линейного пространства не образуют, потому что если $a_{n+1} = -b_{n+1}$, то полином, являющийся их суммой множеству полиномов степени n не принадлежит.

4. Линейное пространство образуют и непрерывные вещественные функции, заданные на промежутке $[a, b]$. Если внутренней и внешней операциями считать обычные операции сложения функций и умножения их на число, то нетрудно убедиться, что это так и есть. Изучение элементов такого пространства является основной задачей математического анализа.

Завершая рассмотрение примеров конкретных линейных пространств, остановимся на вопросе о том, какую роль могут сыграть скаляры из поля **K** при формировании линейного пространства. И внутренняя, и внешняя операции порождают новые элементы, причем внутренняя операция никогда не порождает новые элементы другой природы, чем исходные. Так, например, складывая два вещественных вектора, мы всегда получим вещественный вектор. С внешней операцией дело обстоит иначе. Если в качестве элементов линейного пространства выбраны вещественные векторы, полиномы или функции, то скаляры поля **K** могут быть лишь целыми, рациональными или вещественными. Если скаляры принадлежат комплексному полю, то мультипликативная внешняя операция породит новый элемент, не принадлежащий исходному множеству вещественных векторов. Все сказанное относится ко всем рассмотренным примерам. Например, множество рациональных векторов над полем вещественных или комплексных скаляров линейного пространства не образуют.

Приведенные примеры не исчерпывают всего многообразия конкретных линейных пространств. Следующим шагом будет установление между элементами пространства отношений, которые играют исключительно важную роль при рассмотрении линейных пространств.

Начнем с того, что образуем так называемую линейную комбинацию векторов

$$\alpha\mathbf{x} + \beta\mathbf{y} + \gamma\mathbf{z} + \ldots + \theta\mathbf{v}, \qquad (\Pi2.1)$$

где α, β, γ, ..., θ — скаляры из поля $\mathbf{K}$, а $\mathbf{x}$, $\mathbf{y}$, $\mathbf{z}$, ..., $\mathbf{v}$ — векторы линейного пространства $\mathbf{R}$.

Определение. Векторы $\mathbf{x}$, $\mathbf{y}$, $\mathbf{z}$, ..., $\mathbf{v}$ называются *линейно зависимыми*, если существуют такие скаляры α, β, γ, ..., θ, из которых хотя бы один отличен от нуля, что линейная комбинация (П2.1) равна $\mathbf{0}$.

$$\alpha\mathbf{x} + \beta\mathbf{y} + \gamma\mathbf{z} + \ldots + \theta\mathbf{v} = \mathbf{0}.$$

В противном случае векторы $\mathbf{x}$, $\mathbf{y}$, $\mathbf{z}$, ..., $\mathbf{v}$ называются *линейно независимыми*. Другими словами, векторы $\mathbf{x}$, $\mathbf{y}$, $\mathbf{z}$, ..., $\mathbf{v}$ называются линейно независимыми, если линейная комбинация (П2.1) равна нулю лишь при $\alpha = \beta = \gamma = \ldots = \theta = 0$.

Пусть векторы линейно зависимы, и, например, коэффициент α отличен от нуля. Тогда

$$\alpha\mathbf{x} = -\beta\mathbf{y} - \gamma\mathbf{z} \ldots - \theta\mathbf{v} \text{ или } \mathbf{x} = -\frac{\beta}{\alpha}\mathbf{y} - \frac{\gamma}{\alpha}\mathbf{z} - \ldots - \frac{\theta}{\alpha}\mathbf{v}.$$

Таким образом, вектор $\mathbf{x}$ оказался выраженным через векторы $\mathbf{y}$, $\mathbf{z}$, ..., $\mathbf{v}$. При этом говорят, что вектор $\mathbf{x}$ является линейной комбинацией векторов $\mathbf{y}$, $\mathbf{z}$, ..., $\mathbf{v}$ или что вектор $\mathbf{x}$ разложен по векторам $\mathbf{y}$, $\mathbf{z}$, ..., $\mathbf{v}$.

С понятием линейной зависимости/независимости векторов непосредственно связано понятие размерности линейного пространства. Обсудим это сначала на простых и известных примерах.

На прямой любой вектор может быть выражен через другой вектор (рис. П2.1), т. е. два любых вектора пропорциональны или линейно зависимы.

$$\mathbf{y} = \alpha\mathbf{x} \quad \text{или} \quad 1\mathbf{y} - \alpha\mathbf{x} = \mathbf{0}.$$

Такое пространство целесообразно называть одномерным или размерности 1.

x *y*

Рис. П2.1. Вектор **y** можно выразить через вектор **x**

На плоскости всегда найдутся два линейно независимых вектора, а три любых будут линейно зависимы. Это двумерное пространство. В трехмерном физическом пространстве линейно независимых векторов максимум три, а всякие четыре линейно зависимы и т. д.

Таким образом, на прямой, на плоскости и в физическом пространстве максимальное число линейно независимых векторов совпадает с тем, что в геометрии называется числом измерений соответствующего пространства.

А теперь сформулируем несколько определений.

Определение конечномерности. Линейное пространство называется *конечномерным*, если в нем существует такая конечная система векторов, что любой вектор пространства является линейной комбинацией векторов этой системы.

Определение размерности. Линейное пространство имеет *размерность* n (называется n-мерным), если в нем существует n линейно независимых векторов и нет большего числа линейно независимых векторов.

Определение базиса. Совокупность n линейно независимых векторов $\mathbf{e}_1$, $\mathbf{e}_2$, $\mathbf{e}_3$, ..., $\mathbf{e}_n$ n-мерного пространства называется *базисом*.

Определение координат. Если векторы $\mathbf{e}_1$, $\mathbf{e}_2$, $\mathbf{e}_3$, ..., $\mathbf{e}_n$ есть базис в n-мерном пространстве и

$$\mathbf{x} = \zeta_1\mathbf{e}_1 + \zeta_2\mathbf{e}_2 + \zeta_3\mathbf{e}_3 + \ldots + \zeta_n\mathbf{e}_n,$$

то числа ζ_1, ζ_2, ζ_3, ..., ζ_n называются *координатами* вектора $\mathbf{x}$ в базисе $\{\mathbf{e}\} = (\mathbf{e}_1,\ \mathbf{e}_2,\ \mathbf{e}_3,\ \ldots,\ \mathbf{e}_n)$.

Теперь свяжем введенные определения с некоторыми из ранее рассмотренных пространств.

1. Приложение абстрактных математических понятий к таким физическим пространствам, как прямая, плоскость или реальный физический мир, никаких проблем не вызывает. Прямая, очевидно, является одномерным пространством, а плоскость — двумерным. Из базисов на плоскости наиболее привычен прямоугольный или декартов, хотя и являющийся лишь частным случаем. Аналогично обстоит дело и с трехмерным физическим пространством.

2. Линейное пространство упорядоченных последовательностей скаляров:

$$\mathbf{x} = (x_1,\ x_2,\ x_3,\ \ldots,\ x_n).$$

Возьмем n векторов этого пространства:

$$\mathbf{e}_1 = (1,\ 0,\ ...,\ 0),$$
$$\mathbf{e}_2 = (0,\ 1,\ ...,\ 0),$$
$$...$$
$$\mathbf{e}_n = (0,\ 0,\ ...,\ 1).$$

Покажем, что эти векторы образуют базис пространства.

Проверим линейную независимость этих векторов. Пусть

$$\lambda_1\mathbf{e}_1 + \lambda_2\mathbf{e}_2 + ... + \lambda_n\mathbf{e}_n = \mathbf{0}. \qquad (П2.2)$$

Раскроем левую и правую части этого равенства:

$$\lambda_1(1,\ 0,\ ...,\ 0) + \lambda_2(0,\ 1,\ ...,\ 0) + ... + \lambda_n(0,\ 0,\ ...,\ 1) = (0,\ 0,\ ...,\ 0).$$

Преобразуем левую часть, выполнив необходимые операции:

$$(\lambda_1,\ \lambda_2,\ \lambda_3,\ ...,\ \lambda_n) = (0,\ 0,\ 0,\ ...,\ 0).$$

Отсюда следует: $\lambda_1 = \lambda_2 = \lambda_3 = ... = \lambda_n = 0$. А это значит, что исходное равенство (П2.2) возможно лишь при: $\lambda_1 = \lambda_2 = \lambda_3 = ... = \lambda_n = 0$, т. е. векторы $\mathbf{e}_1,\ \mathbf{e}_2,\ \mathbf{e}_3,\ ...,\ \mathbf{e}_n$ — линейно независимы.

Возьмем произвольный вектор $\mathbf{x} = (x_1, x_2, x_3, ..., x_n)$. Запишем очевидное равенство:

$$(x_1, x_2, x_3, ..., x_n) = x_1(1,0,...,0) + x_2(0,1,...,0) + ... + x_n(0,0,...,1),$$

которое означает не что иное, как разложение вектора $\mathbf{x}$ по базису $\{\mathbf{e}\}$ $\mathbf{x} = x_1\mathbf{e}_1 + x_2\mathbf{e}_2 + x_3\mathbf{e}_3 + ... + x_n\mathbf{e}_n$, где $x_1,\ x_2,\ x_3,\ ...,\ x_n$ — координаты вектора $\mathbf{x}$.

Мы доказали, что $\{\mathbf{e}\}$ образует базис. Размерность пространства равна n.

3. Линейное пространство полиномов с вещественными коэффициентами степени не выше n: $p(t) = a_1 + a_2 t + a_3 t^2 + ... + a_{n+1} t^n$. Базисом этого пространства является, например, такой набор полиномов: $\mathbf{e}_1 = 1$, $\mathbf{e}_2 = t$, $\mathbf{e}_3 = t^2$, ..., $\mathbf{e}_{n+1} = t^n$. Убедимся, что они образуют базис.

Проверим линейную независимость векторов базиса. Так как равенство $\lambda_1 \cdot 1 + \lambda_2 \cdot t + \lambda_3 \cdot t^2 + ... + \lambda_{n+1} \cdot t^n = 0$ означает равенство нулю всех значе-

ний полинома, то это возможно лишь при $\lambda_1 = \lambda_2 = \lambda_3 = ... = \lambda_{n+1} = 0$. Возьмем произвольный полином $p(t) = a_1 + a_2 t + a_3 t^2 + ... + a_{n+1} t^n$.

Ясно, что его можно переписать так: $p(t) = a_1 \mathbf{e}_1 + a_2 \mathbf{e}_2 + a_3 \mathbf{e}_3 + ... + a_{n+1} \mathbf{e}_{n+1}$. То есть произвольный вектор пространства полиномов с вещественными коэффициентами степени не выше n разложен по базису $\{\mathbf{e}\}$, и, значит, a_1, a_2, ..., a_{n+1} — его координаты. Размерность пространства $n+1$.

4. Линейное пространство вещественных непрерывных функций на промежутке $[a, b]$. Пусть N — произвольное целое число. Тогда функции $f_1(t) = 1$, $f_2(t) = t$, $f_3(t) = t^2$, $f_4(t) = t^3$, ..., $f_N(t) = t^{N-1}$ образуют совокупность из N линейно независимых векторов. (Это доказано в предыдущем примере.) Однако, т. к. N — произвольное число, в пространстве может быть произвольное число линейно независимых векторов, т. е. пространство бесконечномерное. Это означает, что при рассмотрении линейного пространства непрерывных на $[a, b]$ вещественных функций, надо иметь в виду, что в нем справедливы лишь те факты конечномерных пространств, доказательство которых не зависит от размерности. К таким фактам относятся аксиомы скалярного произведения и неравенство Коши — Буняковского, понятие ортогональности (об этом речь будет идти позже) и некоторые другие.

Продолжим знакомство с линейными пространствами и рассмотрим переход от одного базиса к другому. Рассмотрим два базиса: "старый" $\{\mathbf{u}\} = (\mathbf{u}_1, \mathbf{u}_2, ..., \mathbf{u}_n)$ и "новый" $\{\mathbf{v}\} = (\mathbf{v}_1, \mathbf{v}_2, ..., \mathbf{v}_n)$. Произвольный вектор $\mathbf{x}$ в базисе $\{\mathbf{u}\}$ выглядит следующим образом:

$$\mathbf{x} = x_1 \mathbf{u}_1 + x_2 \mathbf{u}_2 + x_2 \mathbf{u}_3 + ... + x_n \mathbf{u}_n,$$

а в базисе $\{\mathbf{u}\}$:

$$\mathbf{x} = \bar{x}_1 \mathbf{v}_1 + \bar{x}_2 \mathbf{v}_2 + \bar{x}_3 \mathbf{v}_3 + ... + \bar{x}_n \mathbf{v}_n.$$

Здесь x_1, x_2, x_3, ..., x_n — координаты вектора $\mathbf{x}$ в базисе $\{\mathbf{u}\}$, а $\bar{x}_1$, $\bar{x}_2$, $\bar{x}_3$, ..., $\bar{x}_n$ — координаты того же вектора в базисе $\{\mathbf{v}\}$. Нужно установить связь между "новыми" и "старыми" координатами вектора $\mathbf{x}$.

Для этого запишем разложение векторов "нового" базиса $\mathbf{v}_1$, $\mathbf{v}_2$, ..., $\mathbf{v}_n$ по "старому" базису:

$$\mathbf{v}_1 = a_{11}\mathbf{u}_1 + a_{21}\mathbf{u}_2 + \ldots + a_{n1}\mathbf{u}_n,$$
$$\mathbf{v}_2 = a_{12}\mathbf{u}_1 + a_{22}\mathbf{u}_2 + \ldots + a_{n2}\mathbf{u}_n,$$
$$\ldots$$
$$\mathbf{v}_n = a_{1n}\mathbf{u}_1 + a_{2n}\mathbf{u}_2 + \ldots + a_{nn}\mathbf{u}_n.$$

Здесь числа a_{ij} — координаты векторов $\{\mathbf{v}\}$ в базисе $\{\mathbf{u}\}$. Подставим эти разложения в выражение для $\mathbf{x}$:

$$\mathbf{x} = \bar{x}_1\left(a_{11}\mathbf{u}_1 + a_{21}\mathbf{u}_2 + \ldots + a_{n1}\mathbf{u}_n\right) + \bar{x}_2\left(a_{12}\mathbf{u}_1 + a_{22}\mathbf{u}_2 + \ldots + a_{n2}\mathbf{u}_n\right) + \ldots +$$
$$+ \bar{x}_n\left(a_{1n}\mathbf{u}_1 + a_{2n}\mathbf{u}_2 + \ldots + a_{nn}\mathbf{u}_n\right) =$$
$$= \left(a_{11}\bar{x}_1 + a_{12}\bar{x}_2 + \ldots + a_{1n}\bar{x}_n\right)\mathbf{u}_1 + \left(a_{21}\bar{x}_1 + a_{22}\bar{x}_2 + \ldots + a_{2n}\bar{x}_n\right)\mathbf{u}_2 + \ldots +$$
$$+ \left(a_{n1}\bar{x}_1 + a_{n2}\bar{x}_2 + \ldots + a_{nn}\bar{x}_n\right)\mathbf{u}_n.$$

Получилось другое разложение вектора $\mathbf{x}$ по тем же самым базисным векторам, т. е. в последнем выражении в скобках стоят "старые" координаты, выраженные через "новые". Иными словами:

$$x_1 = a_{11}\bar{x}_1 + a_{12}\bar{x}_2 + \ldots + a_{1n}\bar{x}_n,$$
$$x_2 = a_{21}\bar{x}_1 + a_{22}\bar{x}_2 + \ldots + a_{2n}\bar{x}_n,$$
$$\ldots$$
$$x_n = a_{n1}\bar{x}_1 + a_{n2}\bar{x}_2 + \ldots + a_{nn}\bar{x}_n.$$

Всю совокупность координат a_{ij} удобно записывать в виде квадратной таблицы:

$$\begin{pmatrix} \mathrm{a}_{11} & a_{12} & \ldots & \ldots & a_{1m} \\ a_{21} & a_{22} & \ldots & \ldots & a_{2m} \\ & & \ldots & \ldots & \\ & & \ldots & \ldots & \\ a_{n1} & a_{n2} & \ldots & \ldots & a_{nm} \end{pmatrix}.$$

Такая таблица называется *матрицей*. Выполненные операции можно было проделать и в другом порядке, разлагая векторы "старого" базиса по векторам "нового". Получатся аналогичные формулы, но матрица будет другая. Обе матрицы оказываются связанными, и вторая называется *обратной* по отношению к первой. Следует также заметить, что множество матриц со стандартными операциями сложения и умножения на скаляр образуют линейное пространство.

Все сведения, о которых шла речь до сих пор, определяют аффинные свойства линейных пространств (аффинный — от лат. *affinis* — родственный), которые определяются лишь заданными операциями с элементами. Аффинные преобразования пространства — это такие, при которых сохраняются координаты всех точек при переходе от одного базиса к другому и с помощью которых определяется, что такое прямая, плоскость, параллельные прямые, размерность и т. д.

Мы рассмотрели лишь качественные понятия, игнорируя количественные характеристики, присущие физическим векторам, такие как длина вектора и угол между векторами. Эти характеристики связаны с измерениями, и поэтому наделение пространства ими делает его метрическим.

Внесение в линейное пространство "метрики" или его "метризация", т. е. введение в нем упомянутых характеристик, можно осуществить различными способами. Мы сделаем это аксиоматическим определением скалярного произведения векторов.

Определение. Если каждой паре векторов $\mathbf{x}$ и $\mathbf{y}$ линейного пространства $\mathbf{R}$ поставлен в соответствие скаляр из поля $\mathbf{K}$, обозначаемый $(\mathbf{x}, \mathbf{y})$, то будем говорить, что в линейном пространстве введено скалярное произведение, причем скалярное произведение удовлетворяет аксиомам, представленным в табл. П2.2.

Таблица П2.2. *Аксиомы, которым удовлетворяет скалярное произведение векторов*

Вещественное пространство	**Комплексное пространство**
Симметричность	
1. $(\mathbf{x},\mathbf{y})=(\mathbf{y},\mathbf{x})$	1. $(\mathbf{x},\mathbf{y})=\overline{(\mathbf{y},\mathbf{x})}$
Выносимость скалярного множителя	
2. $(\alpha\mathbf{x},\mathbf{y})=\alpha(\mathbf{x},\mathbf{y})$	2. $(\mathbf{x},\alpha\mathbf{y})=\bar{\alpha}(\mathbf{x},\mathbf{y})$
Дистрибутивность	
3. $(\mathbf{x}+\mathbf{y},\mathbf{z})=(\mathbf{x},\mathbf{z})+(\mathbf{y},\mathbf{z})$	3. $(\mathbf{x}+\mathbf{y},\mathbf{z})=(\mathbf{x},\mathbf{z})+(\mathbf{y},\mathbf{z})$
Положительность	
4. $(\mathbf{x},\mathbf{x})\geq 0$, $(\mathbf{x},\mathbf{x})=0$ (лишь при $\mathbf{x}=0$)	4. $(\mathbf{x},\mathbf{x})\geq 0$, $(\mathbf{x},\mathbf{x})=0$ (лишь при $\mathbf{x}=0$)

Символ " $\overline{\ }$ " означает комплексную сопряженность, обладающую следующими свойствами: $\overline{\mathbf{x}+\mathbf{y}}=\overline{\mathbf{x}}+\overline{\mathbf{y}}$, $\overline{\mathbf{x}\mathbf{y}}=\overline{\mathbf{x}}\cdot\overline{\mathbf{y}}$, $\overline{\overline{\mathbf{x}}}=\mathbf{x}$.

Линейное вещественное пространство с введенным в нем скалярным произведением называется *евклидовым* пространством, а внесенная метрика — *евклидовой*. Линейное комплексное пространство с введенным в нем скалярным произведением называется *унитарным* пространством, а внесенная метрика — *эрмитовой*.

Со скалярным произведением связано знаменитое неравенство Коши — Буняковского — Шварца. Его формулировку определяет следующая теорема.

Теорема. Для любых двух векторов со скалярным произведением справедливо неравенство

$$|(\mathbf{x},\mathbf{y})|^2 \le (\mathbf{x},\mathbf{x})\cdot(\mathbf{y},\mathbf{y}).$$

Доказательство неравенства проведем лишь для вещественного пространства. Цепочка равенств демонстрирует последовательное применение аксиом скалярного произведения:

$$(\alpha\mathbf{x}-\beta\mathbf{y},\alpha\mathbf{x}-\beta\mathbf{y})=(\alpha\mathbf{x}-\beta\mathbf{y},\alpha\mathbf{x})+(\alpha\mathbf{x}-\beta\mathbf{y},-\beta\mathbf{y})=$$
$$=\alpha^2(\mathbf{x},\mathbf{x})-2\alpha\beta(\mathbf{x},\mathbf{y})+\beta^2(\mathbf{y},\mathbf{y})\ge 0.$$

Выберем $\alpha=(\mathbf{x},\mathbf{y})$, $\beta=(\mathbf{x},\mathbf{x})$. Тогда:

$$(\mathbf{x},\mathbf{y})^2(\mathbf{x},\mathbf{x})-2(\mathbf{x},\mathbf{y})^2(\mathbf{x},\mathbf{x})+(\mathbf{x},\mathbf{x})^2(\mathbf{y},\mathbf{y})\ge 0 \;\Rightarrow\; (\mathbf{x},\mathbf{y})^2\le(\mathbf{x},\mathbf{x})(\mathbf{y},\mathbf{y}),$$

что и требовалось доказать.

Рекомендуется самостоятельно выполнить доказательство для комплексного пространства и доказать, что равенство достигается, если **x** и **y** линейно зависимы.

Рассмотрим теперь, как выглядит скалярное произведение в конкретных линейных пространствах, рассмотренных ранее.

1. Проще всего дело обстоит с обычным физическим пространством. Там скалярное произведение вводится как произведение длин векторов на косинус угла между ними, т. е., например, на плоскости:

$$(\mathbf{x},\mathbf{y})=|\mathbf{x}|\cdot|\mathbf{y}|\cdot\cos\theta,$$

где θ — угол между векторами $\mathbf{x}$ и $\mathbf{y}$.

2. Линейное пространство упорядоченных последовательностей вещественных чисел становится евклидовым, если скалярное произведение определяется следующим образом:

$$(\mathbf{x},\mathbf{y}) = x_1 y_1 + x_2 y_2 + \ldots + x_n y_n = \sum_{k=1}^{n} x_k y_k .$$

Для унитарного пространства ситуация несколько иная:

$$(\mathbf{x},\mathbf{y}) = x_1 \overline{y}_1 + x_2 \overline{y}_2 + \ldots + x_n \overline{y}_n = \sum_{k=1}^{n} x_k \overline{y}_k .$$

Заметим, что рассмотренный способ введения скалярного произведения не является единственным.

3. В пространстве полиномов, заданных на $[a,\ b]$, степени не выше n и с вещественными коэффициентами скалярное произведение вводится следующим образом:

$$(p(t), q(t)) = \int_a^b p(t) q(t) dt.$$

4. В пространстве непрерывных функций, заданных на $[a,\ b]$, скалярное произведение вводится аналогично тому, как это делается для полиномов:

$$(f(t), g(t)) = \int_a^b f(t) g(t) dt.$$

Внося метрику линейного пространства посредством введения скалярного произведения, мы оставили в стороне связь таких физических величин, как длина и расстояние, с самим скалярным произведением. Нагляднее и легче это сделать на примере двумерного пространства, т. к. здесь теорема Пифагора и тригонометрические формулы решают все проблемы. При этом линейные пространства упорядоченных последовательностей скаляров являются естественным абстрагированием и обобщением физического пространства.

Запишем формулы метрических соотношений:

$$|\mathbf{x}| = \sqrt{x_1^2 + x_2^2}, \quad |\mathbf{y}| = \sqrt{y_1^2 + y_2^2}, \quad |\mathbf{x} - \mathbf{y}| = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2};$$

$$\cos(\alpha - \beta) = \cos\alpha \cdot \cos\beta + \sin\alpha \cdot \sin\beta =$$

$$= \frac{x_1}{\sqrt{x_1^2 + x_2^2}} \cdot \frac{y_1}{\sqrt{y_1^2 + y_2^2}} + \frac{x_2}{\sqrt{x_1^2 + x_2^2}} \cdot \frac{y_2}{\sqrt{y_1^2 + y_2^2}} = \frac{x_1 y_1 + x_2 y_2}{|\mathbf{x}| \cdot |\mathbf{y}|}.$$

Составными частями выписанных формул являются скалярные произведения:

$$x_1y_1 + x_2y_2 = (\mathbf{x},\mathbf{y}); \quad x_1x_1 + x_2x_2 = (\mathbf{x},\mathbf{x}); \quad y_1y_1 + y_2y_2 = (\mathbf{y},\mathbf{y});$$

$$(x_1 - y_1)^2 + (x_2 - y_2)^2 = (\mathbf{x}-\mathbf{y},\mathbf{x}-\mathbf{y}).$$

В таких обозначениях формулы приобретают вид:

$$|\mathbf{x}| = \sqrt{(\mathbf{x},\mathbf{x})}, \ |\mathbf{y}| = \sqrt{(\mathbf{y},\mathbf{y})}, \ |\mathbf{x}-\mathbf{y}| = \sqrt{(\mathbf{x}-\mathbf{y},\mathbf{x}-\mathbf{y})}, \ \cos(\alpha-\beta) = \frac{(\mathbf{x},\mathbf{y})}{\sqrt{(\mathbf{x},\mathbf{x})}\sqrt{(\mathbf{y},\mathbf{y})}}.$$

Приведенная иллюстрация (рис. П2.2) не может служить доказательством справедливости формул и для n-мерного линейного пространства. С этим можно ознакомиться в многочисленной элементарной и специальной литературе по линейным пространствам.

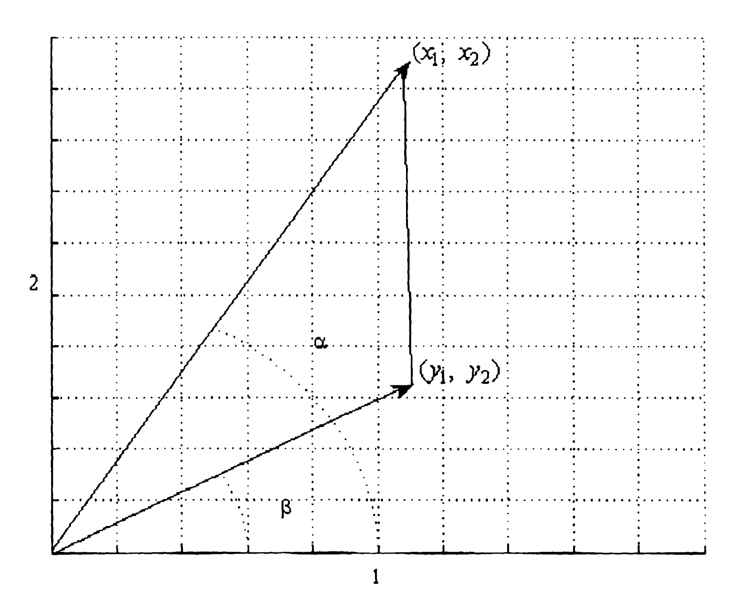


Рис. П2.2. Соотношения между векторами **x** и **y**

Посредством скалярного произведения вводится еще одно важное понятие — ортогональность векторов.

Определение. Векторы, скалярное произведение которых равно нулю, называются *ортогональными*:

$$(\mathbf{x},\mathbf{y}) = 0.$$

Из формулы для косинуса угла между векторами видно, что понятие ортогональности обобщает понятие перпендикулярности:

$$\cos(\alpha-\beta)=\frac{(\mathbf{x},\mathbf{y})}{\sqrt{(\mathbf{x},\mathbf{x})}\sqrt{(\mathbf{y},\mathbf{y})}}.$$

Важной процедурой является ортогонализация заданной системы линейно независимых векторов линейного пространства. Проведение такой операции называют процессом Грама — Шмидта, который состоит в последовательном построении системы ортогональных векторов $\mathbf{v}_1$, $\mathbf{v}_2$, ..., $\mathbf{v}_n$ по системе линейно независимых $\mathbf{u}_1$, $\mathbf{u}_2$, ..., $\mathbf{u}_n$.

Процесс ортогонализации можно было бы сформулировать в виде теоремы и строго доказать ее, пользуясь методом математической индукции. Оставим это на самостоятельную работу и ограничимся лишь описанием процесса ортогонализации.

В качестве первого вектора ортогональной системы $\mathbf{v}_1$ выберем $\mathbf{u}_1$. Второй вектор $\mathbf{v}_2$ будем строить в виде $\mathbf{v}_2=\mathbf{u}_2-\alpha_{21}\mathbf{v}_1$ и таким, чтобы он был ортогонален $\mathbf{v}_1$, т. е. $(\mathbf{v}_2,\ \mathbf{v}_1)=0$. Умножим $\mathbf{v}_2$ скалярно на $\mathbf{v}_1$:

$$(\mathbf{v}_2,\mathbf{v}_1)=0=(\mathbf{u}_2,\mathbf{v}_1)-\alpha_{21}(\mathbf{v}_1,\mathbf{v}_1).$$

Отсюда $\alpha_{21}=\dfrac{(\mathbf{u}_2,\mathbf{v}_1)}{(\mathbf{v}_1,\mathbf{v}_1)}$. Таким образом, вектор $\mathbf{v}_2$ построен.

Далее предположим, что ортогональные отличные от нуля векторы $\mathbf{v}_1$, $\mathbf{v}_2$, ..., $\mathbf{v}_{k-1}$ построены. Вектор $\mathbf{v}_k$ ищем в виде:

$$\mathbf{v}_k=\mathbf{u}_k-\alpha_{k1}\mathbf{v}_1-\alpha_{k2}\mathbf{v}_2-\ldots-\alpha_{k,k-1}\mathbf{v}_{k-1}.$$

Выполнив условия ортогональности вектора $\mathbf{v}_k$ векторам $\mathbf{v}_1$, $\mathbf{v}_2$, ..., $\mathbf{v}_{k-1}$, получим:

$$(\mathbf{v}_k,\mathbf{v}_1)=0=(\mathbf{u}_k,\mathbf{v}_1)-\alpha_{k1}(\mathbf{v}_1,\mathbf{v}_1)-\alpha_{k2}(\mathbf{v}_2,\mathbf{v}_1)-\ldots-\alpha_{k,k-1}(\mathbf{v}_{k-1},\mathbf{v}_1),$$

$$(\mathbf{v}_k,\mathbf{v}_2)=0=(\mathbf{u}_k,\mathbf{v}_2)-\alpha_{k1}(\mathbf{v}_1,\mathbf{v}_2)-\alpha_{k2}(\mathbf{v}_2,\mathbf{v}_2)-\ldots-\alpha_{k,k-1}(\mathbf{v}_{k-1},\mathbf{v}_2),$$

$$\ldots$$

$$(\mathbf{v}_k,\mathbf{v}_{k-1})=0=(\mathbf{u}_k,\mathbf{v}_{k-1})-\alpha_{k1}(\mathbf{v}_1,\mathbf{v}_{k-1})-\alpha_{k2}(\mathbf{v}_2,\mathbf{v}_{k-1})-\ldots-\alpha_{k,k-1}(\mathbf{v}_{k-1},\mathbf{v}_{k-1}).$$

Из этих равенств находим:

$$\alpha_{k1}=\frac{(\mathbf{u}_k,\mathbf{v}_1)}{(\mathbf{v}_1,\mathbf{v}_1)},\quad \alpha_{k2}=\frac{(\mathbf{u}_k,\mathbf{v}_2)}{(\mathbf{v}_2,\mathbf{v}_2)},\quad \alpha_{k,k-1}=\frac{(\mathbf{u}_k,\mathbf{v}_{k-1})}{(\mathbf{v}_{k-1},\mathbf{v}_{k-1})}$$

и вектор $\mathbf{v}_k$ оказывается построенным. Продолжая процесс, получим всю совокупность ортогональных векторов $\{\mathbf{v}_k\}$.

Завершим описание процесса доказательством, что вектор $\mathbf{v}_k$ отличен от нуля. Для этого нам понадобится линейная независимость векторов $\mathbf{u}_1$, $\mathbf{u}_2$, ..., $\mathbf{u}_n$, которая до сих пор нигде не использовалась. Вектор $\mathbf{v}_k$ есть линейная комбинация векторов $\mathbf{u}_k$, $\mathbf{v}_1$, $\mathbf{v}_2$, ..., $\mathbf{v}_{k-1}$, вектор $\mathbf{v}_{k-1}$ — линейная комбинация векторов $\mathbf{u}_{k-1}$, $\mathbf{v}_1$, $\mathbf{v}_2$, ..., $\mathbf{v}_{k-2}$ и т. д. Подставим в выражение для $\mathbf{v}_k$ линейные комбинации, определяющие векторы $\mathbf{v}_{k-1}$, $\mathbf{v}_{k-2}$, ..., $\mathbf{v}_1$. В результате получим:

$$\mathbf{v}_k=1\cdot\mathbf{u}_k+\lambda_{k-1}\mathbf{u}_{k-1}+\lambda_{k-2}\mathbf{u}_{k-2}+\ldots+\lambda_1\mathbf{u}_1,$$

где λ_1, λ_2, ..., λ_{k-1} — коэффициенты, получившиеся в результате приведения подобных членов после подстановки. Совокупность векторов $\{\mathbf{u}\}$ линейно независима, и, значит, линейная комбинация векторов $\{\mathbf{u}\}$ может обращаться в ноль лишь при нулевой комбинации коэффициентов. Однако при векторе $\mathbf{u}_k$ стоит коэффициент, равный единице, т. е. вектор $\mathbf{v}_k$ в ноль обратиться не может. Получаем противоречие.

Построенные векторы $\mathbf{v}_1$, $\mathbf{v}_2$, ..., $\mathbf{v}_n$ определяют только направления ортогональных векторов. Часто удобно иметь дело с векторами некоторой фиксированной длины, например, единичной. Приведение векторов к единичной длине, состоящее в вычислении длин векторов и делении компонентов на них, называется нормированием, а ортогональные векторы единичной длины — *ортонормированными*.

К сожалению, процесс ортогонализации численно неустойчив, и возникающая во время его проведения погрешность часто приводит к значительному нарушению ортогональности строящейся системы векторов. Для борьбы с этим явлением используют *доортогонализацию*, которая развивает и продолжает саму ортогонализацию.

Например, "доворот" вектора $\mathbf{v}_2$ до перпендикулярного по отношению к $\mathbf{v}_1$ положения осуществляется итерационным процессом

$$\mathbf{v}_2^{(k)} = \mathbf{v}_2^{(k-1)} - \Delta\alpha_{21}^{(k)}\mathbf{v}_1, \qquad \Delta\alpha_{21}^{(k)} = \frac{\left(\mathbf{v}_2^{(k-1)}, \mathbf{v}_1\right)}{\left(\mathbf{v}_1, \mathbf{v}_1\right)},$$

где $\mathbf{v}_2^{(1)} = \mathbf{v}_2$, $k = 2, 3, 4, \ldots$

Вычисления прекращаются, когда ортогональность обеспечивается с требуемой точностью, и за окончательное значение $\mathbf{v}_2$ принимается $\mathbf{v}_2^{(k)}$.

Аналогично процесс доортогонализации осуществляется и для следующих векторов $\mathbf{u}_3$, $\mathbf{u}_4$, ... Например:

$$\mathbf{v}_3^{(k)} = \mathbf{v}_3^{(k-1)} - \Delta\alpha_{31}^{(k)}\mathbf{v}_1 - \Delta\alpha_{32}^{(k)}\mathbf{v}_2, \qquad \Delta\alpha_{31}^{(k)} = \frac{\left(\mathbf{v}_3^{(k-1)}, \mathbf{v}_1\right)}{\left(\mathbf{v}_1, \mathbf{v}_1\right)}, \quad \Delta\alpha_{32}^{(k)} = \frac{\left(\mathbf{v}_3^{(k-1)}, \mathbf{v}_2\right)}{\left(\mathbf{v}_2, \mathbf{v}_2\right)},$$

где $\mathbf{v}_3^{(1)} = \mathbf{v}_3$, $k = 2, 3, 4, \ldots$

В подтверждение тезиса о численной неустойчивости процесса ортогонализации приведем численный пример. Все вычисления проводятся с пятью знаками точности.

$$\mathbf{u}_1 = (0.63257,\ 0.31256), \quad \mathbf{u}_2 = (0.56154,\ 0.27740).$$

Нормируем векторы. $|\mathbf{u}_1| = 0.70558$, $|\mathbf{u}_2| = 0.62632$,

$$\mathbf{u}_1^{\text{н}} = \frac{\mathbf{u}_1}{|\mathbf{u}_1|} = (0.89652,\ 0.44298), \quad \mathbf{u}_2^{\text{н}} = \frac{\mathbf{u}_2}{|\mathbf{u}_2|} = (0.89657,\ 0.44290).$$

Как видно, векторы почти параллельны, т. е. почти линейно зависимы. Проведем их ортогонализацию:

$$\mathbf{v}_1^{\text{н}} = \mathbf{u}_1^{\text{н}}, \quad \mathbf{v}_2 = \mathbf{u}_2^{\text{н}} - \frac{\left(\mathbf{u}_2^{\text{н}}, \mathbf{v}_1^{\text{н}}\right)}{\left(\mathbf{v}_1^{\text{н}}, \mathbf{v}_1^{\text{н}}\right)}\mathbf{v}_1^{\text{н}} = (0.00004,\ -0.00008).$$

Нормируем $\mathbf{v}_2$: $\mathbf{v}_2^{\text{н}} = \dfrac{\mathbf{v}_2}{|\mathbf{v}_2|} = (0.44721,\ -0.89442)$.

Проверим ортогональность $\mathbf{v}_1^{\text{н}}$ и $\mathbf{v}_2^{\text{н}}$: $(\mathbf{v}_1^{\text{н}},\ \mathbf{v}_2^{\text{н}}) = 0.00427 \neq 0$. Действительно, ортогональность нарушена и довольно существенно. Нетрудно заметить, что

перпендикулярными к $\mathbf{v}_1^{\text{н}}$ являются векторы $\mathbf{v}_2^{01} = (-0.44298,\ 0.89652)$ и $\mathbf{v}_2^{02} = (0.44298,\ -0.89652)$, отличающиеся лишь знаком, и ортогонализация явно не удалась.

Проведем доортогонализацию.

$$\mathbf{v}_2^{(2)} = \mathbf{v}_2^{(1)} - \frac{\left(\mathbf{v}_2^{(1)}, \mathbf{v}_1^{\text{н}}\right)}{\left(\mathbf{v}_1^{\text{н}}, \mathbf{v}_1^{\text{н}}\right)} \mathbf{v}_1^{\text{н}} =$$

$$= (0.44721,\ -0.89442) - (0.00472/0.99998)(0.89652,\ 0.44298) =$$

$$= (0.44298,\ -0.89651).$$

Оказалось, что достаточно одной итерации.

Проводить ортогонализацию векторов можно в любой последовательности. Вместе с тем, следует учитывать, что порядок выбора векторов может оказать существенное влияние на устойчивость процесса ортогонализации. Так, например, на очередном шаге в качестве вектора $\mathbf{u}_k$ можно привлекать тот, который образует наименьшие скалярные произведения со всеми ортогональными векторами $\mathbf{v}_i$, построенными ранее.

Вернемся к вопросу о связи скалярного произведения с метрическими величинами — длинами, расстояниями и углами. Мы установили, что

$$|\mathbf{x}| = \sqrt{(\mathbf{x}, \mathbf{x})}, \quad \cos(\alpha - \beta) = \frac{(\mathbf{x}, \mathbf{y})}{\sqrt{(\mathbf{x}, \mathbf{x})}\sqrt{(\mathbf{y}, \mathbf{y})}}.$$

При естественных обобщениях и переходе к формальным математическим конструкциям часто появляются понятия, обобщающие физические. Сделаем это в отношении длины вектора. В линейных пространствах говорят о норме вектора, что для физического пространства ничем не отличается от длины. Для нормы вводится и свое обозначение: $\|\mathbf{x}\|$. Пусть пока $\|\mathbf{x}\| = \sqrt{(\mathbf{x}, \mathbf{x})}$. Докажем три теоремы, связанные с нормой.

Теорема 1. $\|\mathbf{x}\| \geq 0$, $\|\mathbf{x}\| = 0$, если $\mathbf{x} = \mathbf{0}$.

Это непосредственно следует из положительности скалярного произведения.

Теорема 2. $\|\alpha\mathbf{x}\| = |\alpha| \cdot \|\mathbf{x}\|$.

$$\|\alpha\mathbf{x}\| = \sqrt{(\alpha\mathbf{x}, \alpha\mathbf{x})} = \sqrt{\alpha^2(\mathbf{x}, \mathbf{x})} = |\alpha|\sqrt{(\mathbf{x}, \mathbf{x})} = |\alpha| \cdot \|\mathbf{x}\|.$$

Теорема 3. $\|\mathbf{x}+\mathbf{y}\| \le \|\mathbf{x}\| + \|\mathbf{y}\|$. (Неравенство треугольника.)

Доказательство этой теоремы основано на неравенстве Коши — Буняковского $(\mathbf{x},\mathbf{y})^2 \le (\mathbf{x},\mathbf{x}) \cdot (\mathbf{y},\mathbf{y})$, которое может быть записано так: $(\mathbf{x},\mathbf{y}) \le \|\mathbf{x}\| \cdot \|\mathbf{y}\|$.

$$\|\mathbf{x}+\mathbf{y}\|^2 = (\mathbf{x}+\mathbf{y},\mathbf{x}+\mathbf{y}) = (\mathbf{x},\mathbf{x}) + 2(\mathbf{x},\mathbf{y}) + (\mathbf{y},\mathbf{y}) = \|\mathbf{x}\|^2 + 2(\mathbf{x},\mathbf{y}) + \|\mathbf{y}\|^2 \le$$

$$\le \|\mathbf{x}\|^2 + 2\|\mathbf{x}\| \cdot \|\mathbf{y}\| + \|\mathbf{y}\|^2 = (\|\mathbf{x}\| + \|\mathbf{y}\|)^2;$$

$$\|\mathbf{x}+\mathbf{y}\| \le \|\mathbf{x}\| + \|\mathbf{y}\|.$$

Теперь эти три теоремы можно трактовать как аксиомы, и с их помощью в линейном пространстве постулировать новое понятие — понятие нормы. Аксиоматическое определение подразумевает, что любые числа, удовлетворяющие аксиомам, являются *нормой*. Одно из множеств таких чисел определяется формулой:

$$\|\mathbf{x}\|_p = \left(|x_1|^p + |x_2|^p + \ldots + |x_n|^p\right)^{\frac{1}{p}},$$

где p — целый параметр.

Тем самым в линейном пространстве метрические характеристики могут быть введены по-иному, чем это делалось ранее с помощью скалярного произведения.

Остается лишь привести некоторые конкретные значения числа p, отвечающие наиболее употребительным нормам. При $p = 2$ имеем:

$$\|\mathbf{x}\|_2 = \sqrt{(\mathbf{x},\mathbf{x})}.$$

Два других значения p, а именно 1 и ∞, порождают следующие нормы:

$$\|\mathbf{x}\|_1 = \sum_{i=1}^{n} |x_i|, \qquad \|\mathbf{x}\|_\infty = \max_i |x_i|.$$

Эти нормы имеют собственные имена:

- октаэдрическая $\|\mathbf{x}\|_1 = \sum_{i=1}^{n} |x_i|$;
- сферическая $\|\mathbf{x}\|_2 = \sqrt{(\mathbf{x},\mathbf{x})}$;
- кубическая $\|\mathbf{x}\|_\infty = \max_i |x_i|$.

Названия произошли от геометрических фигур, которые образуются в результате заполнения некоторого объема всевозможными векторами одинаковой длины.

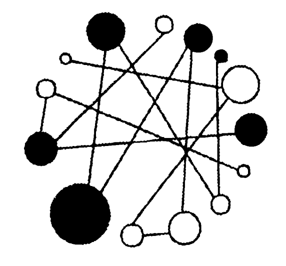
В заключение определим, как выполняются операции дифференцирования и интегрирования с векторами, если их координаты зависят от некоторой независимой переменной t. Тогда

$$\mathbf{x}(t) = \left(x_1(t), x_2(t), \dots, x_n(t)\right), \quad \frac{d\mathbf{x}(t)}{dt} = \left(\frac{dx_1}{dt}, \frac{dx_2}{dt}, \dots, \frac{dx_n}{dt}\right),$$

$$\int_0^t \mathbf{x}(\tau)d\tau = \left(\int_0^t x_1(\tau)d\tau, \int_0^t x_2(\tau)d\tau, \dots, \int_0^t x_n(\tau)d\tau\right),$$

$$\frac{d}{dt}\left(\mathbf{x}(t), \mathbf{y}(t)\right) = \left(\frac{d\mathbf{x}(t)}{dt}, \mathbf{y}(t)\right) + \left(\mathbf{x}(t), \frac{d\mathbf{y}(t)}{dt}\right).$$

ПРИЛОЖЕНИЕ 3



Элементы теории матриц

П3.1. Общие сведения о матрицах

Матрицей называется совокупность $n \times m$ скаляров a_{ij}, образующих прямоугольную таблицу из n строк и m столбцов:

$$\begin{pmatrix} a_{11} & a_{12} & \dots & \dots & a_{1m} \\ a_{21} & a_{22} & \dots & \dots & a_{2m} \\ \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & \dots & \dots & a_{nm} \end{pmatrix}.$$

Скаляры a_{ij} ($i = 1, 2, \dots, n$, $j = 1, 2, \dots, m$) называются *элементами* или *компонентами матрицы*. Первый индекс всегда означает номер строки или число строк, а второй — номер столбца или число столбцов.

Определение матрицы, как единого алгебраического объекта $\mathbf{A}$, принадлежит английскому математику Артуру Кели (Arthur Cayley). Примером математических рассуждений, где появляется такой алгебраический объект, может служить преобразование координат вектора при изменении базиса линейного пространства.

Известные алгебраические объекты, такие как скаляры и векторы, можно считать частными случаями матриц: для скаляров $n = m = 1$, для векторов-строк $n = 1$, для векторов-столбцов $m = 1$. Матрица $\mathbf{0}$ является нулевой, если все ее элементы представлены нулями. При $m = n$ матрица становится квадратной. Элементы a_{11}, a_{22}, a_{33}, ..., a_{nn} образуют главную диагональ, элементы $a_{1,n}$, $a_{2,n-1}$, $a_{3,n-2}$, ..., a_{n1} — побочную. Сумма диагональных элементов образует *след матрицы*:

$$Tr(\mathbf{A}) = \sum_{i=1}^{n} a_{ii}.$$

Если все элементы матрицы, кроме тех, что на главной диагонали, равны нулю, матрица называется *диагональной* и обозначается $\mathbf{D} = \mathrm{diag}(\mathbf{A}) = \mathrm{diag}(a_{11}, a_{22}, ..., a_{nn})$. Диагональная матрица с компонентами, равными единице, является *единичной матрицей* и обозначается буквами **E** или **I**. Если все элементы матрицы выше главной диагонали равны нулю, то такая матрица называется *левой треугольной* или *нижней треугольной*. Аналогично, если все элементы матрицы ниже главной диагонали равны нулю, то такая матрица называется *правой треугольной* или *верхней треугольной*.

Если рассматривать матрицу как сложный объект, состоящий из скаляров, то она может быть подвергнута некоторым преобразованием, в частности, перестановке местами строк и столбцов:

$$\begin{pmatrix} a_{11} & a_{12} & \dots & \dots & a_{1m} \\ a_{21} & a_{22} & \dots & \dots & a_{2m} \\ \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & \dots & \dots & a_{nm} \end{pmatrix} \Rightarrow \begin{pmatrix} a_{11} & a_{21} & \dots & \dots & a_{n1} \\ a_{12} & a_{22} & \dots & \dots & a_{n2} \\ \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots \\ a_{1m} & a_{2m} & \dots & \dots & a_{nm} \end{pmatrix}.$$

Новая матрица называется *транспонированной*, а сама операция — *транспонированием*: $\mathbf{A} \Rightarrow \mathbf{A}^T$. Ясно, что $(\mathbf{A}^T)^T = \mathbf{A}$. Важным классом матриц являются симметрические, а именно те, что не меняются при транспонировании, т. е. когда $\mathbf{A}^T = \mathbf{A}$.

Скаляры, образующие матрицу, могут быть как из множества вещественных, так и из множества комплексных чисел. Как было уже видно, введение в обращение комплексных чисел влечет за собой определенные изменения в метрических характеристиках линейных пространств и их элементов, таких, как скалярное произведение, длина, норма, расстояние между элементами. В комплексном случае операция транспонирования совмещается с заменой комплексных чисел на комплексно-сопряженные, а сама операция называется *эрмитовым сопряжением* и обозначается символами * или ^{H}: $\mathbf{A} \Rightarrow \mathbf{A}^* = \mathbf{A}^H$.

В трехмерном варианте имеем:

$$\begin{pmatrix} a_{11} - ib_{11} & a_{12} - ib_{12} & a_{13} + ib_{13} \\ a_{21} + ib_{21} & a_{22} & a_{23} + ib_{23} \\ a_{31} + ib_{31} & a_{32} - ib_{32} & a_{33} + ib_{33} \end{pmatrix} \Rightarrow \begin{pmatrix} a_{11} + ib_{11} & a_{21} - ib_{21} & a_{31} - ib_{31} \\ a_{12} + ib_{12} & a_{22} & a_{32} + ib_{32} \\ a_{13} - ib_{13} & a_{23} - ib_{23} & a_{33} - b_{33} \end{pmatrix}.$$

Очевидно, что $(\mathbf{A}^H)^H = \mathbf{A}$. Если случится, что $\mathbf{A}^H = \mathbf{A}$, то матрица называется *эрмитовой* или *самосопряженной*.

$$\begin{pmatrix} a_{11} & a_{12}+ib_{12} & a_{13}+ib_{13} \\ a_{21}+ib_{12} & a_{22} & a_{23}+ib_{23} \\ a_{31}+ib_{31} & a_{32}+ib_{32} & a_{33} \end{pmatrix} \Rightarrow \begin{pmatrix} a_{11} & a_{21}-ib_{21} & a_{31}-ib_{31} \\ a_{12}-ib_{12} & a_{22} & a_{32}-ib_{32} \\ a_{13}-ib_{13} & a_{23}-ib_{23} & a_{33} \end{pmatrix}.$$

Из определения следует, что диагональные элементы эрмитовой матрицы вещественны.

П3.2. Операции с матрицами

Матрицы образуют линейное пространство и даже более богатую алгебраическую структуру — кольцо. Это означает, что на множестве матриц, прежде всего, вводятся операции сложения элементов пространства (матриц) и умножения элементов пространства (матриц) на скаляры. Таким образом, порождается линейное пространство. Обе операции определяются покомпонентно, т. е. при сложении складываются элементы с одинаковыми индексами (и это накладывает ограничение на операцию в том смысле, что могут складываться только матрицы с одинаковым числом строк и столбцов), а при умножении на скаляр умножаются все компоненты.

$$\mathbf{A}+\mathbf{B} = \begin{pmatrix} a_{11} & a_{12} & \dots & \dots & a_{1m} \\ a_{21} & a_{22} & \dots & \dots & a_{2m} \\ \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & \dots & \dots & a_{nm} \end{pmatrix} + \begin{pmatrix} b_{11} & b_{12} & \dots & \dots & b_{1m} \\ b_{21} & b_{22} & \dots & \dots & b_{2m} \\ \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots \\ b_{n1} & b_{n2} & \dots & \dots & b_{nm} \end{pmatrix} =$$

$$= \begin{pmatrix} a_{11}+b_{11} & a_{12}+b_{12} & \dots & \dots & a_{1m}+b_{1m} \\ a_{21}+b_{21} & a_{22}+b_{22} & \dots & \dots & a_{2m}+b_{2m} \\ & & \dots & \dots & \\ & & \dots & \dots & \\ a_{n1}+b_{n1} & a_{n2}+b_{n2} & \dots & \dots & a_{nm}+b_{nm} \end{pmatrix},$$

$$\alpha\mathbf{A} = \begin{pmatrix} \alpha a_{11} & \alpha a_{12} & \dots & \dots & \alpha a_{1m} \\ \alpha a_{21} & \alpha a_{22} & \dots & \dots & \alpha a_{2m} \\ & & \dots & \dots & \\ & & \dots & \dots & \\ \alpha a_{n1} & \alpha a_{n2} & \dots & \dots & \alpha a_{nm} \end{pmatrix}.$$

Нейтральным элементом для аддитивной операции являются нулевая матрица, а единицей для умножения матрицы на скаляр — скалярная единица. Обе дистрибутивные операции выглядят как обычно в линейных пространствах:

$$(\alpha + \beta)\mathbf{A} = \alpha\mathbf{A} + \beta\mathbf{A},$$

$$\alpha(\mathbf{A} + \mathbf{B}) = \alpha\mathbf{A} + \alpha\mathbf{B}.$$

Введение мультипликативной операции с матрицами порождает структуру кольца. Естественным является вопрос: почему кольца, а не поля? Это становится ясным после определения самой операции умножения матриц. Компоненты c_{ij} матрицы произведения $\mathbf{C}_{np}$ вычисляются как скалярные произведения строк левой матрицы $\mathbf{A}_{nm}$ на столбцы правой матрицы $\mathbf{B}_{mp}$:

$$c_{ij} = \sum_{k=1}^{m} a_{ik} b_{kj}.$$

Легко убедиться, что введенная операция ассоциативна: $(\mathbf{A} \cdot \mathbf{B}) \cdot \mathbf{C} = \mathbf{A} \cdot (\mathbf{B} \cdot \mathbf{C})$. Но она не коммутативна. Именно поэтому и не получается структура поля. Убедиться в некоммутативности можно, даже не выполняя никаких операций. Например, при умножении матрицы $\mathbf{A}_{24}$ на матрицу $\mathbf{B}_{42}$ получится матрица $\mathbf{C}_{22}$, а при умножении $\mathbf{B}_{42} \cdot \mathbf{A}_{24} = \mathbf{C}_{44}$. Более того, для матриц $\mathbf{A}_{24}$ и $\mathbf{B}_{43}$ вторую матрицу на первую умножать вообще нельзя.

Частным случаем является умножение матрицы на вектор:

$$\mathbf{Ax} = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} a_{11}x_1 + a_{12}x_2 \\ a_{21}x_1 + a_{22}x_2 \end{pmatrix}.$$

Здесь и в дальнейшем будем употреблять термин "вектор" применительно к вектору-столбцу.

Обратимся к аксиомам существования нейтрального и обратного элементов для операции умножения матриц. Прежде всего, заметим, что речь может идти лишь о квадратных матрицах. Роль нейтрального элемента естественно играет единичная матрица: $\mathbf{A} \cdot \mathbf{E} = \mathbf{A}$. Проблема существования обратного элемента в аксиоме $\mathbf{A} \cdot \mathbf{A}^{-1} = \mathbf{E}$ решается следующим образом. Каждой квадратной матрице ставится в соответствие скаляр $\det(\mathbf{A})$, называемый *определителем матрицы*. Определитель есть сумма произведений элементов матрицы, таких, что в каждом произведении n сомножителей и каждая строка и каждый столбец представлены по одному разу. Знак каждого слагаемого оп-

ределяется совпадением (+) или несовпадением (–) четностей перестановок первого и второго индексов элементов. Если определитель оказывается равным нулю, матрицу называют *особенной* или *вырожденной*, в противном случае — *неособенной* или *невырожденной*. Обратный элемент $\mathbf{A}^{-1}$ существует лишь для неособенных матриц. Из множества свойств определителей потребуется следующее: $\det(\mathbf{A}\cdot\mathbf{B}) = \det(\mathbf{A})\cdot\det(\mathbf{B})$.

Нахождение обратной матрицы связано с довольно трудоемкими вычислениями. Однако существуют матрицы, обратные для которых находятся относительно легко. В частности, такими матрицами являются диагональные и треугольные. Особое место занимают матрицы, для которых обратная равна транспонированной, т. е. $\mathbf{Q}^T = \mathbf{Q}^{-1}$, и с учетом равенства $\mathbf{Q}^{-1}\mathbf{Q} = \mathbf{E}$ имеем $\mathbf{Q}^T\mathbf{Q} = \mathbf{E}$. Так как элементы матрицы произведения вычисляются как скалярные произведения строк левой матрицы на столбцы правой, легко заметить, что единичная матрица как результат умножения может получиться тогда и только тогда, когда столбцы матрицы $\mathbf{Q}$ ортонормированы. Такая матрица называется *ортогональной* и обозначается буквой $\mathbf{Q}$.

$$\mathbf{Q}^T\mathbf{Q} = \begin{pmatrix} ----q_1^T---- \\ ----q_2^T---- \\ \dots \\ \dots \\ ----q_n^T---- \end{pmatrix} \begin{pmatrix} | & | & & & | \\ | & | & & & | \\ q_1 & q_2 & \dots & \dots & q_n \\ | & | & & & | \\ | & | & & & | \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & \dots & 0 \\ 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ & & \dots & & \\ 0 & 0 & 0 & & 1 \end{pmatrix}.$$

Ортонормированными оказываются и строки матрицы с учетом равенства $\mathbf{Q}^T\mathbf{Q} = \mathbf{E} = \mathbf{Q}\mathbf{Q}^T$.

Приведем примеры ортогональных матриц, ограничившись матрицами 2×2.

- *Матрица вращений* (поворачивает вектор на угол θ без изменения длины вектора)

$$\mathbf{Q} = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix}, \quad \mathbf{Q}^T = \mathbf{Q}^{-1} = \begin{pmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{pmatrix}.$$

- *Матрица перестановки*

$$\mathbf{P} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad \mathbf{P}^T = \mathbf{P}^{-1} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}.$$

Матрица $\mathbf{P}$ отображает каждую точку (x, y) в ее зеркальный образ (x, y) относительно прямой $y = x$, проходящей под углом 45° к оси x. В трехмерном случае матрица $\mathbf{P}$ имеет вид:

$$\mathbf{P} = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{pmatrix}, \quad \mathbf{P}^T = \mathbf{P}^{-1} = \begin{pmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}.$$

Наиболее важным свойством ортогональных матриц является то, что умножение на ортогональную матрицу сохраняет длины и скалярные произведения:

$$\|\mathbf{Qx}\| = \|\mathbf{x}\| \quad (\mathbf{Qx}, \mathbf{Qy}) = (\mathbf{x}, \mathbf{y}).$$

В случае комплексных элементов матрица с ортонормированными столбцами называется *унитарной* и обозначается буквой $\mathbf{U}$, а основное соотношение имеет вид:

$$\mathbf{U}^H = \mathbf{U}^{-1}.$$

Присутствие заметных различий между алгеброй скаляров и алгеброй матриц иллюстрирует и следующий факт.

Алгебра матриц: *нулевая матрица может быть разложена на ненулевые множители, т. е.* $\mathbf{0} = \mathbf{AB}$, *причем* $\mathbf{A} \neq \mathbf{0}$ *и* $\mathbf{B} \neq \mathbf{0}$.

Алгебра скаляров: *нуль не может быть разложен на ненулевые множители, т. е. если* $\mathbf{ab} = \mathbf{0}$, *то либо* $\mathbf{a} = \mathbf{0}$, *либо* $\mathbf{b} = \mathbf{0}$, *либо* $\mathbf{a} = \mathbf{b} = \mathbf{0}$.

Сформулируем несколько свойств обратной и самосопряженной матриц, доказательство которых рекомендуется провести самостоятельно:

- $(\mathbf{A}^{-1})^{-1} = \mathbf{A}$;
- $(\mathbf{AB})^{-1} = \mathbf{B}^{-1}\mathbf{A}^{-1}$;
- $\det \mathbf{A}^{-1} = 1/\det \mathbf{A}$;
- $(\mathbf{AB})^* = \mathbf{B}^*\mathbf{A}^*$;
- $(\mathbf{A}^{-1})^* = (\mathbf{A}^*)^{-1}$.

Еще несколько задач связано со свойствами диагональных и треугольных матриц.

- При умножении матрицы $\mathbf{A}$ на диагональную матрицу $\mathbf{D}$ *слева* $\mathbf{B} = \mathbf{DA}$ все *строки* $\mathbf{A}$ умножаются на соответствующие диагональные элементы.

- При умножении матрицы **A** на диагональную матрицу **D** справа **B** = **AD** все *столбцы* **A** умножаются на соответствующие диагональные элементы. Как частный случай, имеем равенства **A** = **EA** = **AE**.
- При умножении треугольных матриц одного вида (например, левых треугольных) получается матрица того же вида (т. е. левая треугольная).
- Обратная матрица для левой треугольной является треугольной матрицей того вида.

Введение операций с матрицами позволяет вместо условного обозначения для скалярного произведения в евклидовом и унитарном пространствах использовать формулы, отражающие арифметические операции. Приняв за основу изображение вектора столбцом

$$\mathbf{x} = \begin{pmatrix} x_1 \\ x_2 \\ \\ \\ x_n \end{pmatrix},$$

строку будем записывать $\mathbf{x}^T = (x_1, x_2, ..., x_n)^T$. Тогда скалярное произведение в евклидовом пространстве будет вычисляться по формуле $\mathbf{x}^T\mathbf{y} = \sum_{k=1}^{n} x_k y_k = (\mathbf{x}, \mathbf{y})$, а в унитарном $\mathbf{x}^T\overline{\mathbf{y}} = \mathbf{y}^*\mathbf{x} = \sum_{k=1}^{n} x_k \overline{y_k} = (\mathbf{x}, \mathbf{y})$, где компоненты вектора $\overline{\mathbf{y}}$ сопряжены по отношению к вектору $\mathbf{y}$, а $\mathbf{y}^*$ — вектор, эрмитово сопряженный с вектором $\mathbf{y}$. В этих обозначениях свойство сохранения скалярного произведения для ортогональной матрицы имеет вид: $(\mathbf{Qx})^T(\mathbf{Qy}) = \mathbf{x}^T\mathbf{Q}^T\mathbf{Qy} = \mathbf{x}^T\mathbf{y}$.

Весьма полезное свойство имеет место для скалярного произведения с матрицей $(\mathbf{Ax}, \mathbf{y})$: $(\mathbf{Ax}, \mathbf{y}) = \mathbf{y}^*\mathbf{Ax} = (\mathbf{A}^*\mathbf{y})^*\mathbf{x} = (\mathbf{x}, \mathbf{A}^*\mathbf{y})$. Таким образом, матрицу можно перемещать из одного сомножителя скалярного произведения в другой с выполнением операции сопряжения (в вещественном случае — транспонирования).

Среди всех самосопряженных матриц особо важную роль играют положительно определенные матрицы. Самосопряженная матрица называется положительно определенной, если $(\mathbf{Ax}, \mathbf{x}) > 0$ для всех ненулевых векторов $\mathbf{x}$. Аналогично, с неравенством $(\mathbf{By}, \mathbf{y}) < 0$ связано определение отрицательно

9 Зак. 3418

определенной матрицы. Используя положительно определенную матрицу, можно ввести скалярное произведение с матрицей ($\mathbf{Ax}$, $\mathbf{y}$), удовлетворяющее всем аксиомам обычного скалярного произведения:

- $(\mathbf{Ax},\ \mathbf{y}) = (\mathbf{Ay},\ \mathbf{x})$;
- $(\alpha\mathbf{Ax},\ \mathbf{y}) = \alpha(\mathbf{Ax},\ \mathbf{y})$;
- $(\mathbf{A}(\mathbf{x}+\mathbf{y}),\ \mathbf{z}) = (\mathbf{Ax},\ \mathbf{z}) + (\mathbf{Ay},\ \mathbf{z})$;
- $(\mathbf{Ax},\ \mathbf{x}) \geq 0$; $(\mathbf{Ax},\ \mathbf{x}) = 0$, если $\mathbf{z} = \mathbf{0}$.

Завершим раздел несколькими терминологическими замечаниями. Современное теоретико-множественное представление об отображении, которое каждому элементу одного множества ставит в соответствие элемент другого множества, вобрало в себя частные случаи отображаемых и отображающих множеств, а исторически некоторые отображения имеют свое название. Так, отображение элементов множества континуума (числовой оси) на числовую же ось издавна называлось *функцией*. Например:

$$y = \sin(x), \quad y = a + be^x, \quad y = f(x).$$

Отображение элементов линейного пространства во множество элементов числовой оси называют *функционалом*. Вот примеры:

$$\|\mathbf{x}\|_1 = \sum_{i=1}^{n} |x_i|, \quad \|\mathbf{x}\|_\infty = \max|x_i|, \quad y = \int_a^b f(x)dx.$$

Самым общим отображением является оператор, когда элементы некоторого пространства отображаются на элементы этого же или другого пространства. Оператор обычно обозначают так: $\mathbf{g} = \mathbf{Af}$. Здесь $\mathbf{f}$ и $\mathbf{g}$ — элементы линейного или более общего, например, банахова или гильбертова пространства, а $\mathbf{A}$ — оператор. Оператором является матрица $\mathbf{A}$ в равенстве $\mathbf{y} = \mathbf{Ax}$ (а само равенство представляет собой линейное преобразование), оператор представлен интегралом в выражении $g(x) = \int_a^b K(x,y)f(y)dy$, ставящем в соответствие каждой функции f функцию g.

Таким образом, матрицы в функциональном смысле можно рассматривать как операторы. Такие операторы удовлетворяют условию линейности:

$$\mathbf{A}(\lambda_1\mathbf{x}_1 + \lambda_2\mathbf{x}_2) = \lambda_1\mathbf{Ax}_1 + \lambda_2\mathbf{Ax}_2,$$

где λ_1, λ_2 — скаляры, $\mathbf{x}_1$, $\mathbf{x}_2$ — векторы, $\mathbf{A}$ — оператор (матрица). Равенство это непосредственно основано на матричных операциях, введенных в

(П3.2). В алгебре матриц говорят, что матрица $\mathbf{A}$ есть матрица линейного преобразования.

П3.3. Собственные значения и собственные векторы матриц

Рассмотрим линейное преобразование с квадратной матрицей $\mathbf{A}_{nn}$:

$$\mathbf{y} = \mathbf{A}\mathbf{u}.$$

Поставим задачу: для заданной матрицы $\mathbf{A}$ найти такие векторы $\mathbf{u}$, которые сохраняют свое направление после линейного преобразования. Другими словами, каковы должны быть векторы $\mathbf{u}$, чтобы вектор $\mathbf{y}$ был пропорционален или коллинеарен вектору $\mathbf{u}$, т. е. $\mathbf{y} = \lambda\mathbf{u}$. Тогда

$$\mathbf{A}\mathbf{u} = \lambda\mathbf{u} \text{ или } (\mathbf{A} - \lambda\mathbf{E})\mathbf{u} = \mathbf{0}. \qquad (\text{П3.1})$$

Числа λ и векторы $\mathbf{u}$, удовлетворяющие этому уравнению, получили название *"собственные значения"* и *"собственные векторы"* соответственно. Равенство (П3.1) представляет собой однородную линейную алгебраическую систему относительно искомых компонентов вектора $\mathbf{u}$. Известно, что если определитель матрицы $\mathbf{A} - \lambda\mathbf{E}$ не равен нулю, то система имеет единственное тривиальное нулевое решение $\mathbf{u} = \mathbf{0}$. Чтобы система (П3.1) имела нетривиальные решения, нужно потребовать, чтобы ее определитель был равен нулю: $\det(\mathbf{A} - \lambda\mathbf{E}) = 0$. Запишем это равенство в покомпонентном виде:

$$\det\left(\begin{pmatrix} a_{11} & a_{12} & \dots & \dots & a_{1m} \\ a_{21} & a_{22} & \dots & \dots & a_{2m} \\ & & \dots & \dots & \\ & & \dots & \dots & \\ a_{n1} & a_{n2} & \dots & \dots & a_{nm} \end{pmatrix} - \begin{pmatrix} \lambda & 0 & 0 & \dots & 0 \\ 0 & \lambda & 0 & \dots & 0 \\ 0 & 0 & \lambda & \dots & 0 \\ & & \dots & & \\ 0 & 0 & 0 & & \lambda \end{pmatrix}\right) =$$

$$= \det\begin{pmatrix} a_{11} - \lambda & a_{12} & \dots & \dots & a_{1m} \\ a_{21} & a_{22} - \lambda & \dots & \dots & a_{2m} \\ & & \dots & \dots & \\ & & \dots & \dots & \\ a_{n1} & a_{n2} & \dots & \dots & a_{nm} - \lambda \end{pmatrix} = 0.$$

Вычисление этого определителя дает:

$$\det(\mathbf{A}-\lambda\mathbf{E})=(-1)^n\lambda^n+b_1\lambda^{n-1}+\ldots+b_{n-1}\lambda+b_n=0. \qquad (\Pi 3.2)$$

Полином, стоящий в левой части этого уравнения, называется *характеристическим полиномом*, а само уравнение — *характеристическим уравнением*. Оно имеет ровно n корней с учетом их кратности. Это и есть собственные значения матрицы $\lambda_1,\ \lambda_2,\ \ldots,\ \lambda_n$. Они составляют спектр матрицы $\mathbf{A}$, а величина $\rho(\mathbf{A})=\max\limits_i|\lambda_i|$, $(i=1,2,\ldots,n)$ называется *спектральным радиусом*.

Для каждого λ_i можно найти решение $\mathbf{u}_i$ однородной системы $\mathbf{A}\mathbf{u}_i=\lambda_i\mathbf{u}_i$, т. е. собственный вектор. Если все собственные значения различны, то $\mathbf{u}_i$ образуют систему из n различных векторов.

Поскольку собственные векторы находятся из решения однородной системы, то и известными они оказываются с точностью до постоянного (ненулевого) множителя, т. е. собственные векторы однозначно определены по направлению, но их длины (нормы) остаются произвольными. Часто бывает удобно приводить векторы к единичной длине, т. е. нормировать их.

Приведем пример вычисления собственных значений и собственных векторов матрицы

$$\mathbf{A}=\begin{pmatrix}-51 & -49\\ -50 & -50\end{pmatrix}.$$

Собственные значения:

$$\det(A-\lambda E)=\det\begin{pmatrix}-51-\lambda & -49\\ -50 & -50-\lambda\end{pmatrix}=\lambda^2+101\lambda+100=0,$$

$$\lambda_1=-100,\quad \lambda_2=-1\,.$$

Собственные векторы:

$$\begin{pmatrix}-51 & -49\\ -50 & -50\end{pmatrix}\begin{pmatrix}u_{11}\\ u_{12}\end{pmatrix}=-100\begin{pmatrix}u_{11}\\ u_{12}\end{pmatrix},$$

$$-49u_{12}=-49u_{11}$$

$$u_{12}=u_{11}$$

$$\mathbf{u}_1=\begin{pmatrix}1\\ 1\end{pmatrix}u_{11}$$

$$\begin{pmatrix}-51 & -49\\ -50 & -50\end{pmatrix}\begin{pmatrix}u_{21}\\ u_{22}\end{pmatrix}=-1\begin{pmatrix}u_{21}\\ u_{22}\end{pmatrix}$$

$$-50u_{21}=49u_{22}$$

$$u_{21}=\left(\frac{-49}{50}\right)u_{22}$$

$$\mathbf{u}_2=\begin{pmatrix}\dfrac{-49}{50}\\ 1\end{pmatrix}u_{22}.$$

Нормируем собственные векторы:

$$\|\mathbf{u}_1\| = \sqrt{(\mathbf{u}_1, \mathbf{u}_1)} = \sqrt{2}u_{11} \qquad \|\mathbf{u}_2\| = \sqrt{(\mathbf{u}_2, \mathbf{u}_2)} = \sqrt{\left(\frac{49}{50}\right)^2 + 1}\; u_{21}$$

$$\mathbf{u}_1^{\text{норм}} = \frac{\mathbf{u}_1}{\|\mathbf{u}_1\|} = \frac{1}{\sqrt{2}}\begin{pmatrix}1\\1\end{pmatrix} \qquad \mathbf{u}_2^{\text{норм}} = \frac{\mathbf{u}_2}{\|\mathbf{u}_2\|} = \sqrt{\frac{2500}{4901}}\begin{pmatrix}\frac{-49}{50}\\1\end{pmatrix}.$$

Остановимся на некоторых свойствах собственных значений и собственных векторов.

Теорема 1. Собственные векторы, соответствующие различным собственным значениям, линейно независимы.

Доказательство. Составим линейную комбинацию собственных векторов, соответствующих различным собственным значениям:

$$c_1\mathbf{u}_1 + c_2\mathbf{u}_2 + \ldots + c_k\mathbf{u}_k = \mathbf{0}.$$

Умножим ее на матрицу $(\mathbf{A} - \lambda_1\mathbf{E})$:

$$c_1(\mathbf{A} - \lambda_1\mathbf{E})\mathbf{u}_1 + c_2(\mathbf{A} - \lambda_1\mathbf{E})\mathbf{u}_2 + \ldots + c_k(\mathbf{A} - \lambda_1\mathbf{E})\mathbf{u}_k = \mathbf{0}.$$

Первое слагаемое равно нулю по определению собственного вектора. Оставшуюся часть умножим на матрицу $(\mathbf{A} - \lambda_2\mathbf{E})$:

$$c_2(\mathbf{A} - \lambda_2\mathbf{E})(\mathbf{A} - \lambda_1\mathbf{E})\mathbf{u}_2 + \ldots + c_k(\mathbf{A} - \lambda_2\mathbf{E})(\mathbf{A} - \lambda_1\mathbf{E})\mathbf{u}_k = \mathbf{0}.$$

И в этой сумме первое слагаемое равно нулю, что легко видеть из нижеследующего:

$$c_2(\mathbf{A} - \lambda_2\mathbf{E})(\mathbf{A} - \lambda_1\mathbf{E})\mathbf{u}_2 = c_2(\mathbf{A} - \lambda_2\mathbf{E})(\mathbf{A}\mathbf{u}_2 - \lambda_1\mathbf{u}_2) =$$
$$= c_2(\mathbf{A} - \lambda_2\mathbf{E})(\lambda_2\mathbf{u}_2 - \lambda_1\mathbf{u}_2) = c_2(\lambda_2 - \lambda_1)(\mathbf{A} - \lambda_2\mathbf{E})\mathbf{u}_2.$$

Продолжая описанную процедуру, придем к равенству:

$$c_k(\mathbf{A} - \lambda_{k-1}\mathbf{E})(\mathbf{A} - \lambda_{k-2}\mathbf{E})\ldots(\mathbf{A} - \lambda_1\mathbf{E})\mathbf{u}_2 = \mathbf{0},$$

которое возможно лишь при $c_k = 0$. Теперь, просматривая в обратном порядке все равенства, получавшиеся после умножения на очередную матрицу $(\mathbf{A} - \lambda_i\mathbf{E})$, видим, что $c_{k-1} = 0$, $c_{k-2} = 0$, ..., $c_1 = 0$, что и доказывает линейную независимость собственных векторов $\mathbf{u}_1$, $\mathbf{u}_2$, , ..., $\mathbf{u}_k$.

Следствие. Если все собственные значения матрицы различны, то все собственные векторы — линейно независимы.

Определение. Матрица $\mathbf{A}$ называется *матрицей простой структуры*, если все ее собственные векторы линейно независимы.

Очевидно, что матрица с различными собственными значениями имеет простую структуру. Значительно сложнее обстоит дело с собственными векторами у матриц с кратными собственными значениями. Однако и такие матрицы могут иметь простую структуру. Примером может служить единичная матрица $\mathbf{E}_{nn}$, имеющая n линейно независимых собственных векторов и одно собственное число кратности n, равное 1.

Теорема 2. Собственные значения самосопряженной матрицы вещественны.

Доказательство. Рассмотрим два равенства: $\mathbf{Au} = \lambda\mathbf{u}$ и получившееся из первого после операции сопряжения $\mathbf{u}^*\mathbf{A}^* = \bar{\lambda}\mathbf{u}^*$. Умножим первое слева на $\mathbf{u}^*$, а второе справа на $\mathbf{u}$ и вычтем второе из первого:

$$\mathbf{u}^*\mathbf{Au} - \mathbf{u}^*\mathbf{A}^*\mathbf{u} = \lambda\mathbf{u}^*\mathbf{u} - \bar{\lambda}\mathbf{u}^*\mathbf{u}.$$

Выражение слева равно нулю, т. к. $\mathbf{A}^* = \mathbf{A}$. Тогда $(\lambda - \bar{\lambda})\mathbf{u}^*\mathbf{u} = 0$. Скалярное произведение $\mathbf{u}^*\mathbf{u} \neq 0$, значит, $\lambda = \bar{\lambda}$, что возможно лишь для вещественных чисел.

Теорема 3. Собственные векторы самосопряженной матрицы, соответствующие различным собственным значениям, ортогональны.

Доказательство. Возьмем два различных собственных значения λ_i и λ_k и соответствующие им собственные векторы $\mathbf{u}_i$ и $\mathbf{u}_k$. Равенство $\mathbf{Au}_i = \lambda\mathbf{u}_i$ скалярно умножим на $\mathbf{u}_k$, а равенство $\mathbf{Au}_k = \lambda\mathbf{u}_k$ на $\mathbf{u}_i$:

$$(\mathbf{Au}_i, \mathbf{u}_k) = \lambda_i(\mathbf{u}_i, \mathbf{u}_k);\ (\mathbf{Au}_k, \mathbf{u}_i) = \lambda_k(\mathbf{u}_k, \mathbf{u}_i).$$

В первом равенстве переместим $\mathbf{A}$ во второй сомножитель с учетом, что $\mathbf{A}^* = \mathbf{A}$, и вычтем получившееся равенство из второго:

$$(\mathbf{Au}_k, \mathbf{u}_i) - (\mathbf{u}_i, \mathbf{Au}_k) = \lambda_k(\mathbf{u}_k, \mathbf{u}_i) - \lambda_i(\mathbf{u}_i, \mathbf{u}_k).$$

В итоге имеем: $0 = (\lambda_k - \lambda_i)(\mathbf{u}_k, \mathbf{u}_i)$. По условию $\lambda_k - \lambda_i \neq 0$. Следовательно, $(\mathbf{u}_k, \mathbf{u}_i) = 0$, что и означает ортогональность собственных векторов $\mathbf{u}_k$ и $\mathbf{u}_i$.

Представление о свойствах матрицы $\mathbf{A}$ будет более полным, если рассмотреть собственные векторы транспонированной матрицы $\mathbf{A}^T$ или сопряженной $\mathbf{A}^*$. Определитель матрицы не изменяет своего значения, если строки и столбцы меняются местами, поэтому характеристические полиномы и уравнения для матриц $\mathbf{A}$ и $\mathbf{A}^T$ совпадают, т. е. $\det(\mathbf{A}-\lambda\mathbf{E})=\det\left(\mathbf{A}^T-\lambda\mathbf{E}\right)=0$. Следовательно, и собственные значения обеих матриц будут одни и те же. (Для комплексных матриц собственные значения и коэффициенты характеристических полиномов матриц $\mathbf{A}$ и $\mathbf{A}^*$ комплексно сопряжены.) Однако собственные векторы у $\mathbf{A}$ и $\mathbf{A}^T$, вообще говоря, будут различными. Таким образом, есть n собственных значений λ_1, λ_2, ..., λ_n и $2n$ собственных векторов, а именно n собственных векторов матрицы $\mathbf{A}$: $\mathbf{u}_1$, $\mathbf{u}_2$, ..., $\mathbf{u}_n$ и n собственных векторов матрицы $\mathbf{A}^T$: $\mathbf{v}_1,\mathbf{v}_2,...,\mathbf{v}_n$.

Уравнение для собственных векторов матрицы $\mathbf{A}^T$

$$\mathbf{A}^T\mathbf{v}_i=\lambda_i\mathbf{v}_i$$

можно преобразовать с помощью операции транспонирования в

$$\mathbf{v}_i^T\mathbf{A}=\lambda_i\mathbf{v}_i^T,$$

и поэтому иногда векторы $\mathbf{v}_i$ называют *левыми собственными векторами* матрицы $\mathbf{A}$, а $\mathbf{u}_i$ — *правыми*.

Между правыми и левыми собственными векторами матрицы $\mathbf{A}$ может быть установлена связь. Умножим равенство $\mathbf{v}_i^T\mathbf{A}=\lambda_i\mathbf{v}_i^T$ справа на $\mathbf{u}_k$, а равенство $\mathbf{A}\mathbf{u}_k=\lambda_k\mathbf{u}_k$ слева на $\mathbf{v}_i^T$ и вычтем одно из другого

$$0=(\lambda_i-\lambda_k)\mathbf{v}_i^T\mathbf{u}_k,$$

т. е. $\mathbf{v}_i^T\mathbf{u}_k=0$ при $\lambda_i\neq\lambda_k$.

Если векторы имеют вещественные элементы, то $\mathbf{v}_i^T\mathbf{u}_k=\mathbf{u}_k^T\mathbf{v}_i=(\mathbf{u}_k,\mathbf{v}_i)=0$ для $i\neq k$. Это значит, что каждый вектор из одного ряда ортогонален любому вектору из другого, за исключением вектора с тем же индексом. Такое свойство называется биортогональностью, а векторы *биортогональными*. Для векторов с комплексными компонентами (т. е. в унитарном пространстве) $\mathbf{v}_i^T\mathbf{u}_k$ не является скалярным произведением.

Скалярное произведение $\mathbf{v}_k^T\mathbf{u}_k$ из-за неопределенности длин собственных векторов может принимать произвольное значение. Часто удобно считать, что $\mathbf{v}_k^T\mathbf{u}_k = 1$. Добиться этого можно нормированием скалярного произведения по одному из векторов, например по $\mathbf{u}_k$. Если $\mathbf{v}_k^T\mathbf{u}_k = c_k$, то, заменив $\mathbf{u}_k$ на $\tilde{\mathbf{u}}_k = \mathbf{u}_k / c_{k\,i}$, получим $\mathbf{v}_k^T\tilde{\mathbf{u}}_k = 1$.

Произвольный вектор $\mathbf{x}$ может быть разложен по собственным векторам матрицы $\mathbf{u}_1$, $\mathbf{u}_2$, ..., $\mathbf{u}_n$:

$$\mathbf{x} = \beta_1\mathbf{u}_1 + \beta_1\mathbf{u}_2 + \ldots + \beta_n\mathbf{u}_n.$$

Умножим вектор $\mathbf{x}$ справа на вектор $\mathbf{v}_k^T$:

$$\mathbf{v}_k^T\mathbf{x} = \beta_1\mathbf{v}_k^T\mathbf{u}_1 + \beta_1\mathbf{v}_k^T\mathbf{u}_2 + \ldots + \beta_n\mathbf{v}_k^T\mathbf{u}_n.$$

Так как $\mathbf{v}_k^T\mathbf{u}_i = 0$ для $i \neq k$, то

$$\beta_k = \frac{\mathbf{v}_k^T\mathbf{x}}{\mathbf{v}_k^T\mathbf{u}_k},\ k = 1, 2, \ldots, n$$

или $\beta_k = \mathbf{v}_k^T\mathbf{x}$, $\mathbf{v}_k^T\tilde{\mathbf{u}}_k = 1$.

Теперь вектор $\mathbf{x}$ можно представить следующим образом:

$$\mathbf{x} = \sum_{k=1}^{n}\frac{\mathbf{v}_k^T\mathbf{x}}{\mathbf{v}_k^T\mathbf{u}_k}\mathbf{u}_k = \sum_{k=1}^{n}\left(\mathbf{v}_k^T\mathbf{x}\right)\tilde{\mathbf{u}}_k.$$

Если элементы векторов являются вещественными, то в этой формуле под знаком суммы можно записать скалярные произведения:

$$\mathbf{x} = \sum_{k=1}^{n}\frac{(\mathbf{v}_k, \mathbf{x})}{(\mathbf{v}_k, \mathbf{u}_k)}\mathbf{u}_k.$$

Продолжим пример, рассмотренный в начале раздела, и найдем левые собственные векторы матрицы, а потом убедимся в их биортогональности с уже найденными правыми.

$$\mathbf{A} = \begin{pmatrix} -51 & -49 \\ -50 & -50 \end{pmatrix}.$$

Собственные значения: $\lambda_1 = -100$, $\lambda_2 = -1$.

Левые собственные векторы:

$$\begin{pmatrix} -51 & -50 \\ -49 & -50 \end{pmatrix}\begin{pmatrix} v_{11} \\ v_{12} \end{pmatrix} = -100\begin{pmatrix} v_{11} \\ v_{12} \end{pmatrix}, \qquad \begin{pmatrix} -51 & -50 \\ -49 & -50 \end{pmatrix}\begin{pmatrix} v_{21} \\ v_{22} \end{pmatrix} = -1\begin{pmatrix} v_{21} \\ v_{22} \end{pmatrix}$$

$$-50v_{12} = -49v_{11} \qquad -50v_{22} = 50v_{21}$$

$$v_{12} = \frac{49}{50}v_{11} \qquad -v_{21} = v_{22}$$

$$\mathbf{v}_1 = \begin{pmatrix} 1 \\ \frac{49}{50} \end{pmatrix} v_{11} \qquad \mathbf{v}_2 = \begin{pmatrix} -1 \\ 1 \end{pmatrix} v_{22}.$$

Правые уже были вычислены: $\mathbf{u}_1 = \begin{pmatrix} 1 \\ 1 \end{pmatrix} u_{11}$, $\mathbf{u}_2 = \begin{pmatrix} -\frac{49}{50} \\ 1 \end{pmatrix} u_{22}$.

Проверяем биортогональность:

$$(\mathbf{v}_1, \mathbf{u}_2) = -1\frac{49}{50} + \frac{49}{50}\cdot 1 = 0 \qquad (\mathbf{v}_2, \mathbf{u}_1) = -1\cdot 1 + 1\cdot 1 = 0$$

$$(\mathbf{u}_1, \mathbf{v}_1) = 1 + \frac{49}{50} = \frac{99}{50} \qquad (\mathbf{u}_2, \mathbf{v}_2) = \frac{49}{50} + 1 = \frac{99}{50}.$$

Нормируем скалярные произведения по одному из векторов, например по $\mathbf{v}$:

$$\tilde{\mathbf{v}}_1 = \frac{50}{99}\mathbf{v}_1 = \begin{pmatrix} \frac{50}{99} \\ \frac{49}{99} \end{pmatrix}, \qquad \tilde{\mathbf{v}}_2 = \frac{50}{99}\mathbf{v}_2 = \begin{pmatrix} -\frac{50}{99} \\ \frac{50}{99} \end{pmatrix}.$$

Проверяем ортонормированность:

$$(\mathbf{u}_1, \tilde{\mathbf{v}}_1) = \frac{50}{99} + \frac{49}{99} = 1, \qquad (\mathbf{u}_2, \tilde{\mathbf{v}}_2) = \frac{49}{99} + \frac{50}{99} = 1.$$

В заключение данного раздела предлагается самостоятельно убедиться в справедливости следующих утверждений:

- собственные значения диагональной матрицы равны ее диагональным элементам;
- собственные значения треугольной матрицы равны ее диагональным элементам;

- сумма всех собственных значений произвольной квадратной матрицы $\mathbf{A}$ равна сумме ее диагональных элементов (следу матрицы $\mathrm{Tr}(\mathbf{A})$), а произведение всех собственных значений равно ее определителю $\det(\mathbf{A})$.

П3.4. Нормы матриц

В конце *приложения 2* каждому вектору уже ставился в соответствие скаляр, называемый *нормой* и характеризующий в некотором смысле "величину" вектора. Представляется уместным ввести аналогичную характеристику и для матрицы. Естественно сохранить обозначение $\|\mathbf{A}\|$ и название — норма матрицы $\mathbf{A}$. Введем это понятие аксиоматически. Пусть α — некоторый скаляр. *Нормой матрицы* $\|\mathbf{A}\|$ называется число, удовлетворяющее следующим четырем аксиомам:

- $\|\mathbf{A}\| \geq 0$, $\|\mathbf{A}\| = 0$ тогда и только тогда, когда $\mathbf{A} = \mathbf{0}$;
- $\|\alpha\mathbf{A}\| = |\alpha|\|\mathbf{A}\|$;
- $\|\mathbf{A} + \mathbf{B}\| \leq \|\mathbf{A}\| + \|\mathbf{B}\|$;
- $\|\mathbf{A} \cdot \mathbf{B}\| \leq \|\mathbf{A}\| \cdot \|\mathbf{B}\|$.

Если дополнительно выполняются еще две аксиомы, то такая норма называется *канонической*:

- $|a_{ik}| \leq \|\mathbf{A}\|; \quad \forall i,k$;
- если $\forall i,k \quad |a_{ik}| \leq |b_{ik}|$, то $\|\mathbf{A}\| \leq \|\mathbf{B}\|$.

Норма матрицы может быть введена различными способами. Во многих приложениях матрицы присутствуют одновременно с векторами. Поэтому целесообразно вводить норму матрицы так, чтобы она была разумным образом связана с используемой в данном случае нормой вектора. Так матричная норма $\|\mathbf{A}\|_c$ называется *согласованной* с данной нормой вектора, если для любой матрицы $\mathbf{A}$ и любого вектора $\mathbf{x}$ справедливо неравенство:

$$\|\mathbf{A}\mathbf{x}\| \leq \|\mathbf{A}\|_c \cdot \|\mathbf{x}\|. \tag{П3.3}$$

Весьма актуальной представляется задача построения наименьшей нормы, согласованной с данной нормой вектора. Возьмем некоторую норму вектора

$\|\mathbf{x}\|$ и всевозможные векторы, имеющие единичную длину: $\|\mathbf{x}\|=1$. На множестве таких векторов рассмотрим значения нормы вектора $\mathbf{Ax}$: $\|\mathbf{Ax}\|$. Среди величин $\|\mathbf{Ax}\|$ в силу непрерывности нормы обязательно найдется максимальная. Она и принимается за норму матрицы $\mathbf{A}$. Определенная таким образом норма матрицы называется *подчиненной* векторной норме.

$$\|\mathbf{A}\|_i = \sup_{\|x\|=1} \|\mathbf{Ax}\|. \qquad (П3.4)$$

Оказывается, что эта норма удовлетворяет всем четырем аксиомам нормы и условию согласованности (П3.3).

То, что требование согласованности выполняется, устанавливается следующим образом. Пусть $\mathbf{x} \neq \mathbf{0}$. Нормируем вектор $\mathbf{x}$ и рассмотрим вектор $\mathbf{x}^{\text{норм}} = \dfrac{\mathbf{x}}{\|\mathbf{x}\|}$ с единичной нормой. Вычислим $\|\mathbf{Ax}\|$ с учетом того, что $\mathbf{x} = \|\mathbf{x}\| \cdot \mathbf{x}^{\text{норм}}$:

$$\|\mathbf{Ax}\| = \left\|\mathbf{A}\left(\|\mathbf{x}\| \cdot \mathbf{x}^{\text{норм}}\right)\right\| = \left\|\mathbf{Ax}^{\text{норм}}\right\| \cdot \|\mathbf{x}\| \le \left(\sup\left\|\mathbf{Ax}^{\text{норм}}\right\|\right) \cdot \|\mathbf{x}\| = \|\mathbf{A}\| \cdot \|\mathbf{x}\|.$$

Если $\mathbf{x} \neq \mathbf{0}$, то справедливость условия согласованности очевидна. Таким образом, векторная норма и подчиненная ей матричная нормы всегда согласованы.

Теорема. Норма матрицы, подчиненная данной норме вектора, не больше любой согласованной с той же нормой вектора: $\|\mathbf{A}\|_n \le \|\mathbf{A}\|_c$.

Доказательство. В силу непрерывности нормы всегда найдется вектор $\mathbf{x}_0$, такой, что $\|\mathbf{x}_0\|=1$ и $\|\mathbf{Ax}_0\| = \|\mathbf{A}\|_n$. Но $\|\mathbf{Ax}_0\| \le \|\mathbf{A}\|_c \cdot \|\mathbf{x}_0\| = \|\mathbf{A}\|_c$, и, значит $\|\mathbf{A}\|_n \le \|\mathbf{A}\|_c$.

Можно показать, что для наиболее употребительных векторных норм

$$\|\mathbf{x}\|_1 = \sum_{i=1}^{n} |x_i|, \quad \|\mathbf{x}\|_2 = (\mathbf{x}, \mathbf{x})^{1/2}, \quad \|\mathbf{x}\|_\infty = \max_i |x_i|$$

подчиненными являются следующие матричные нормы, обозначаемые теми же индексами, что и векторные:

$$\|\mathbf{A}\|_1 = \max_j \sum_{i=1}^{n} |a_{ij}|, \quad \|\mathbf{A}\|_2 = \left(\rho\left(\mathbf{A}^*\mathbf{A}\right)\right)^{1/2}, \quad \|\mathbf{A}\|_\infty = \max_i \sum_{j=1}^{n} |a_{ij}|.$$

Исторически сложилось так, что в качестве норм используется много различных чисел. Так существует вторая важная норма, согласованная со сферической нормой вектора $\|\mathbf{x}\|_2 = (\mathbf{x}, \mathbf{x})^{1/2}$. Это $\|\mathbf{A}\|_E = \left(\sum_{i=1}^{n} \sum_{j=1}^{n} |a_{ij}|^2 \right)^{1/2}$ — евклидова норма. Не трудно показать, что $\|\mathbf{A}\|_1$, $\|\mathbf{A}\|_\infty$, $\|\mathbf{A}\|_E$ — это не просто нормы, а нормы канонические. Евклидова норма не может быть подчиненной никакой векторной норме, т. к. $\|\mathbf{E}\|_E = \sqrt{n}$. Достаточно популярная *М*-норма $\|\mathbf{A}\|_M = n \cdot \max |a_{ij}|$ согласована со всеми тремя нормами вектора $\|\mathbf{x}\|_1$, $\|\mathbf{x}\|_2$, $\|\mathbf{x}\|_\infty$.

В качестве примера приведем доказательство согласованности одной из матричных норм с векторной, а именно $\|\mathbf{A}\|_\infty$ с $\|\mathbf{x}\|_\infty$.

$$\|\mathbf{x}\|_\infty = \max_i |x_i|, \qquad \|\mathbf{A}\|_\infty = \max_i \sum_{j=1}^{n} |a_{ij}|;$$

$$\|\mathbf{Ax}\|_\infty = \max_i \left| \sum_{j=1}^{n} a_{ij} x_j \right| \le \max_i \sum_{j=1}^{n} |a_{ij} x_j| \le \max_i \sum_{j=1}^{n} |a_{ij}| \cdot |x_j| \le$$

$$\le \max_i \sum_{j=1}^{n} |a_{ij}| \cdot \max_j |x_j| = \|\mathbf{A}\|_\infty \cdot \|\mathbf{x}\|_\infty.$$

Доказательство согласованности других норм осуществляется аналогично.

П3.5. Матричный ряд и матричные функции

В данном разделе речь пойдет об отображении, где областью определения функции является множество матриц, которые, как известно, образуют линейное пространство и не только линейное пространство. При этом в качестве области значений выступает множество матриц. Возможно, более корректным термином в данном случае был бы "оператор". Однако современная литература предпочитает слово "функция", по-видимому, вследствие большого внешнего сходства матричных функций со скалярными.

Наиболее простыми матричными функциями являются полиномы. Для построения полинома необходимо образовать степени матрицы $\mathbf{A}$: $\mathbf{A}^0$, $\mathbf{A}^1$,

$\mathbf{A}^2$, ..., $\mathbf{A}^n$. Степени матриц определены лишь для квадратных матриц, и поэтому в дальнейшем речь будет идти только о них. По определению $\mathbf{A}^k = \mathbf{AAA...A}$ (матрица умножается k раз) и $\mathbf{A}^0 = \mathbf{E}$. Тогда матричный полином со скалярными коэффициентами c_0, c_1, ..., c_n будет выглядеть так:

$$\mathbf{P}_n(\mathbf{A}) = c_0\mathbf{E} + c_1\mathbf{A} + c_2\mathbf{A}^2 + ... + c_n\mathbf{A}^n .$$

Аргументом (область определения) является квадратная матрица размера $m \times m$, и значением будет матрица (область значений) того же размера.

Теперь устремим n к бесконечности, т. е. формально перейдем к бесконечной сумме

$$\mathbf{P}(\mathbf{A}) = \sum_{\gamma=0}^{\infty} c_\gamma \mathbf{A}^\gamma. \qquad (П3.5)$$

Такая сумма называется *степенным матричным рядом* относительно матрицы $\mathbf{A}$. Матричному ряду естественно сопоставить скалярный ряд

$$p(x) = \sum_{\gamma=0}^{\infty} c_\gamma x^\gamma .$$

Матричный ряд будем называть сходящимся, если сходятся все m^2 скалярных рядов для элементов матрицы $\mathbf{P}(\mathbf{A})$. Введенное нами понятие нормы позволяет установить достаточное условие сходимости матричного ряда. Введем матрицу $\mathbf{U}^{(\gamma)} = c_\gamma \mathbf{A}^\gamma$. Обозначим ее элементы за $u_{kj}^{(\gamma)}$, а элементы матрицы $\mathbf{P}(\mathbf{A})$ за $p_{k,j}$. Тогда с учетом выполнения шести аксиом для канонической нормы имеем цепочку неравенств:

$$\begin{aligned} \left|p_{k,j}\right| = \left|\sum_{\gamma=0}^{\infty} u_{kj}^{(\gamma)}\right| \le \sum_{\gamma=0}^{\infty}\left|u_{kj}^{(\gamma)}\right| \le \sum_{\gamma=0}^{\infty}\left\|c_\gamma \mathbf{A}^\gamma\right\| = \\ = \sum_{\gamma=0}^{\infty}\left|c_\gamma\right|\left\|\mathbf{A}^\gamma\right\| \le \sum_{\gamma=0}^{\infty}\left|c_\gamma\right|\left\|\mathbf{A}\right\|^\gamma. \end{aligned} \qquad (П3.6)$$

В результате *достаточным* условием сходимости матричного ряда (П3.5) является выполнение условия

$$\left\|\mathbf{A}\right\| < R, \qquad (П3.7)$$

являющегося, в свою очередь, условием абсолютной сходимости скалярного степенного ряда, стоящего последним в цепочке (П3.6). Здесь R — радиус сходимости скалярного степенного ряда.

Правильному пониманию условия (П3.7) способствует ответ на следующие два вопроса. Пусть имеется ряд с $R = 17$. Сходится ли матричный ряд (П3.5), если

- $\|\mathbf{A}\|_1 = 24, \quad \|\mathbf{A}\|_\infty = 18, \quad \|\mathbf{A}\|_E = 16;$
- $\|\mathbf{A}\|_1 = 24, \quad \|\mathbf{A}\|_\infty = 18, \quad \|\mathbf{A}\|_E = 17.5$?

Необходимое условие сходимости матричного ряда рассмотрим несколько позже.

Если матричный ряд сходится, то матрицу $\mathbf{P}(\mathbf{A})$ будем называть *матричной функцией* (пока ограничимся матричными функциями только такого вида). Примерами матричных функций могут служить

$$e^{\mathbf{A}} = \sum_{k=0}^{\infty} \frac{\mathbf{A}^k}{k!}, \quad \cos(\mathbf{A}) = \sum_{k=0}^{\infty} \frac{(-1)^k \mathbf{A}^{2k}}{(2k)!},$$

$$\sin(\mathbf{A}) = \sum_{k=0}^{\infty} \frac{(-1)^k \mathbf{A}^{2k+1}}{(2k+1)!}, \quad (\mathbf{E} - \mathbf{A})^{-1} = \sum_{k=0}^{\infty} \mathbf{A}^k$$

Первые три ряда (матричные экспонента, косинус и синус), так же, как и их скалярные аналоги, сходятся для любых матриц, а последний ряд (геометрическая прогрессия) имеет $R = 1$. Попутно легко убедиться, что $\sin^2(\mathbf{A}) + \cos^2(\mathbf{A}) = \mathbf{E}$.

Для дальнейшего изложения необходимо ввести понятие подобных матриц.

Определение. Пусть задана матрица $\mathbf{A}$ и некоторая неособенная матрица $\mathbf{S}$ (т. е. $\det(\mathbf{S}) \neq 0$ и существует $\mathbf{S}^{-1}$). Всякая матрица $\mathbf{B} = \mathbf{SAS}^{-1}$ называется *подобной* матрице $\mathbf{A}$. Очевидно, что и $\mathbf{A} = \mathbf{S}^{-1}\mathbf{BS}$ подобна $\mathbf{B}$.

С точки зрения линейных преобразований можно сказать, что две матрицы подобны, если они соответствуют одному и тому же линейному преобразованию в различных базисах. Поясним, что это означает. В некотором базисе вектор $\mathbf{x}$ преобразуется в вектор $\mathbf{y}$ посредством матрицы $\mathbf{A}$: $\mathbf{y} = \mathbf{Ax}$. Переход к новому базису осуществляет матрица $\mathbf{S}$, т. е. образы векторов $\mathbf{x}$ и $\mathbf{y}$ в новом базисе имеют вид: $\xi = \mathbf{Sx}$ и $\upsilon = \mathbf{Sy}$. Умножив оба равенства слева на

$\mathbf{S}^{-1}$, получим $\mathbf{x}=\mathbf{S}^{-1}\xi$ и $\mathbf{y}=\mathbf{S}^{-1}\upsilon$. В новом базисе равенство $\mathbf{y}=\mathbf{A}\mathbf{x}$ превращается в $\mathbf{S}^{-1}\upsilon=\mathbf{A}\mathbf{S}^{-1}\xi$ или $\upsilon=\mathbf{S}\mathbf{A}\mathbf{S}^{-1}\xi$. Это и означает, что матрица $\mathbf{S}\mathbf{A}\mathbf{S}^{-1}$ (подобная матрице $\mathbf{A}$) осуществляет то же самое линейное преобразование, что и $\mathbf{A}$, но в другом базисе.

Сформулируем теперь несколько теорем о подобных матрицах и матричных функциях, устанавливающих некоторые свойства тех и других.

Теорема 1. Подобные матрицы $\mathbf{A}$ и $\mathbf{B}=\mathbf{S}\mathbf{A}\mathbf{S}^{-1}$ имеют одинаковые собственные значения. При этом, если собственному значению λ матрицы $\mathbf{A}$ отвечает собственный вектор $\mathbf{u}$, то у матрицы $\mathbf{B}$ этому же собственному числу λ соответствует собственный вектор $\mathbf{S}\mathbf{u}$.

Доказательство. Так как $\mathbf{S}\mathbf{S}^{-1}=\mathbf{E}$, то $\det(\mathbf{S}\mathbf{S}^{-1}) = \det(\mathbf{S})\det(\mathbf{S}^{-1}) = \det(\mathbf{E}) = 1$. Для характеристических полиномов $\mathbf{A}$ и $\mathbf{B}$ имеем:

$$\det(\mathbf{B}-\lambda\mathbf{E})=\det\left(\mathbf{S}\mathbf{A}\mathbf{S}^{-1}-\lambda\mathbf{S}\mathbf{S}^{-1}\right)=\det\left(\mathbf{S}(\mathbf{A}-\lambda\mathbf{E})\mathbf{S}^{-1}\right)=$$
$$=\det(\mathbf{S})\cdot\det(\mathbf{A}-\lambda\mathbf{E})\cdot\det(\mathbf{S}^{-1})=\det(\mathbf{A}-\lambda\mathbf{E}).$$

Характеристические полиномы для обеих матриц совпали, следовательно, совпали и их корни, т. е. собственные значения. Для доказательства второй части теоремы в равенстве $\mathbf{A}\mathbf{u}=\lambda\mathbf{u}$ заменим матрицу $\mathbf{A}$ на подобную ей $\mathbf{S}^{-1}\mathbf{B}\mathbf{S}$:

$$\mathbf{S}^{-1}\mathbf{B}\mathbf{S}\mathbf{u}=\lambda\mathbf{u}.$$

Теперь, умножив обе части равенства на $\mathbf{S}$ слева, получим требуемый результат:

$$\mathbf{B}(\mathbf{S}\mathbf{u})=\lambda(\mathbf{S}\mathbf{u}).$$

Теорема 2. Если матрицы $\mathbf{A}$ и $\mathbf{B}$ подобны, то их матричные функции также подобны. Иными словами, если $\mathbf{B}=\mathbf{S}\mathbf{A}\mathbf{S}^{-1}$, то $\mathbf{f}(\mathbf{B})=\mathbf{S}\mathbf{f}(\mathbf{A})\mathbf{S}^{-1}$.

Доказательство. Первоначально определим $\mathbf{B}^k$.

$$\mathbf{B}^k=\left(\mathbf{S}\mathbf{A}\mathbf{S}^{-1}\right)^k=\mathbf{S}\mathbf{A}\mathbf{S}^{-1}\mathbf{S}\mathbf{A}\mathbf{S}^{-1}...\mathbf{S}\mathbf{A}\mathbf{S}^{-1}=\mathbf{S}\mathbf{A}\mathbf{A}...\mathbf{A}\mathbf{S}^{-1}=\mathbf{S}\mathbf{A}^k\mathbf{S}^{-1},$$

$$\mathbf{f}(\mathbf{B})=\sum_{k=0}^{\infty}c_k\mathbf{B}^k=\mathbf{S}\left(\sum_{k=0}^{\infty}c_k\mathbf{A}^k\right)\mathbf{S}^{-1}=\mathbf{S}\mathbf{f}(\mathbf{A})\mathbf{S}^{-1}.$$

Теорема 3. Матрица $\mathbf{A}$ простой структуры с собственными значениями λ_1, λ_2, ..., λ_m подобна некоторой диагональной матрице $\mathbf{\Lambda}$, на главной диагонали которой стоят собственные значения матрицы $\mathbf{A}$, т. е. $\mathbf{\Lambda} = \mathrm{diag}(\lambda_1, \lambda_2, ..., \lambda_m)$.

Доказательство. Пусть $\mathbf{u}_k$ — собственные векторы матрицы $\mathbf{A}$. Обозначим за $\mathbf{U}$ матрицу, столбцами которой являются все $\mathbf{u}_k$. Тогда

$$\mathbf{AU} = \mathbf{A}\begin{pmatrix} | & | & & | \\ | & | & & | \\ \mathbf{u}_1 & \mathbf{u}_2 & \dots\ \dots & \mathbf{u}_m \\ | & | & & | \\ | & | & & | \end{pmatrix} = \begin{pmatrix} | & | & & | \\ | & | & & | \\ \lambda_1\mathbf{u}_1 & \lambda_2\mathbf{u}_2 & \dots\ \dots & \lambda_m\mathbf{u}_m \\ | & | & & | \\ | & | & & | \end{pmatrix} =$$

$$= \begin{pmatrix} | & | & & | \\ | & | & & | \\ \mathbf{u}_1 & \mathbf{u}_2 & \dots\ \dots & \mathbf{u}_m \\ | & | & & | \\ | & | & & | \end{pmatrix} \cdot \begin{pmatrix} \lambda_1 & 0 & 0 & \dots & 0 \\ 0 & \lambda_2 & 0 & \dots & 0 \\ 0 & 0 & \lambda_3 & \dots & 0 \\ & & \dots & & \\ 0 & 0 & 0 & & \lambda_m \end{pmatrix} = \mathbf{U\Lambda}.$$

Умножая полученное равенство поочередно справа и слева на $\mathbf{U}^{-1}$, получаем требуемое

$$\mathbf{\Lambda} = \mathbf{U}^{-1}\mathbf{AU}, \quad \mathbf{A} = \mathbf{U\Lambda U}^{-1}.$$

Попутно заметим, что мы одновременно не только доказали теорему, но и определили матрицу преобразования подобия $\mathbf{U}$, состоящую из линейно независимых столбцов $\mathbf{u}_k$ и, следовательно, неособенную. Не вдаваясь в подробности, отметим, что в общем случае, при наличии кратных собственных значений у матрицы $\mathbf{A}$, уже не имеющей простую структуру, вместо матрицы $\mathbf{\Lambda}$ возникает клеточно-диагональная матрица, где каждая клетка представляет собой так называемый *канонический ящик Жордана.*

Исключительно для простоты изложения дальнейшие теоремы будут доказываться только для матриц простой структуры, однако результаты справедливы и для более общего случая.

Теорема 4. Если собственные значения матрицы $\mathbf{A}$ обозначить через λ_1, λ_2, ..., λ_m, то собственными значениями матрицы $\mathbf{f}(\mathbf{A})$ будут числа $f(\lambda_1)$, $f(\lambda_2)$, ..., $f(\lambda_m)$.

Доказательство. $\mathbf{A} = \mathbf{U\Lambda U}^{-1}$. По теореме 2 $\mathbf{f}(\mathbf{A}) = \mathbf{U}^{-1}\mathbf{f}(\mathbf{\Lambda})\mathbf{U}$. Представим $\mathbf{f}(\mathbf{\Lambda})$ в покомпонентном виде

$$\mathbf{f}(\mathbf{\Lambda}) = \sum_{k=0}^{\infty} c_k \mathbf{\Lambda}^k = \begin{pmatrix} \sum_{k=0}^{\infty} c_k\lambda_1^k & 0 & . . & 0 \\ 0 & \sum_{k=0}^{\infty} c_k\lambda_2^k & . . & 0 \\ . & . & . . & . \\ . & . & . . & . \\ 0 & 0 & . . & \sum_{k=0}^{\infty} c_k\lambda_m^k \end{pmatrix} =$$

$$= \begin{pmatrix} f(\lambda_1) & 0 & . . & 0 \\ 0 & f(\lambda_2) & . . & 0 \\ . & . & . . & . \\ . & . & . . & . \\ 0 & 0 & . . & f(\lambda_m) \end{pmatrix}.$$

И тогда

$$\mathbf{f}(\mathbf{A}) = \mathbf{U}^{-1} f(\mathbf{\Lambda})\mathbf{U} = \mathbf{U}^{-1} \begin{pmatrix} f(\lambda_1) & 0 & . . & 0 \\ 0 & f(\lambda_2) & . . & 0 \\ . & . & . . & . \\ . & . & . . & . \\ 0 & 0 & . . & f(\lambda_m) \end{pmatrix} \mathbf{U},$$

что и означает, что $f(\lambda_1)$, $f(\lambda_2)$, ..., $f(\lambda_m)$ являются собственными значениями матрицы $\mathbf{f}(\mathbf{A})$, т. к. преобразование подобия не меняет собственных значений.

Следствие 1. Из вышеприведенных формул непосредственно следует, что матричный ряд $\mathbf{f}(\mathbf{A})$ существует тогда и только тогда, когда существуют *все* скалярные степенные ряды, стоящие на диагонали матрицы $\mathbf{f}(\mathbf{\Lambda})$, а у тех, в свою очередь, необходимым и достаточным условием существования является выполнение условий

$$|\lambda_k| < R, \quad \forall \lambda_k. \qquad (\text{П}3.8)$$

10 Зак. 3418

Таким образом, условие (П3.8) является необходимым и достаточным условием сходимости матричного степенного ряда. На практике вопрос о сходимости матричного ряда решается в такой последовательности:

- сначала находится радиус сходимости R соответствующего скалярного ряда;
- затем проверяется выполнение условия (П3.8) для всех собственных значений.

Следствие 2. Поскольку условие (П3.7) является лишь достаточным условием сходимости матричного степенного ряда, а условие (П3.8) необходимым и достаточным, то из совместного рассмотрения обоих условий легко заключить, что

$$|\lambda_k| \le \|\mathbf{A}\|, \tag{П3.9}$$

т. е. что все собственные значения матрицы не превышают ее любую каноническую норму. Формула (П3.9) будет многократно использоваться нами в дальнейшем. В ее справедливости рекомендуется убедиться самостоятельно.

Теорема 5. Две любые функции матрицы $\mathbf{A}$ коммутируют между собой:

$$\mathbf{f}(\mathbf{A})\cdot\mathbf{g}(\mathbf{A}) = \mathbf{g}(\mathbf{A})\cdot\mathbf{f}(\mathbf{A}).$$

Доказательство. По теореме 2

$$\mathbf{f}(\mathbf{A}) = \mathbf{U}^{-1}\mathbf{f}(\boldsymbol{\Lambda})\mathbf{U} \text{ и } \mathbf{g}(\mathbf{A}) = \mathbf{U}^{-1}\mathbf{g}(\boldsymbol{\Lambda})\mathbf{U}.$$

В силу того, что диагональные матрицы всегда коммутируют, имеем:

$$\mathbf{f}(\mathbf{A})\mathbf{g}(\mathbf{A}) = \mathbf{U}^{-1}\mathbf{f}(\boldsymbol{\Lambda})\mathbf{U}\mathbf{U}^{-1}\mathbf{g}(\boldsymbol{\Lambda})\mathbf{U} = \mathbf{U}^{-1}\mathbf{f}(\boldsymbol{\Lambda})\mathbf{g}(\boldsymbol{\Lambda})\mathbf{U} =$$

$$= \mathbf{U}^{-1}\mathbf{g}(\boldsymbol{\Lambda})\mathbf{f}(\boldsymbol{\Lambda})\mathbf{U} = \mathbf{U}^{-1}\mathbf{g}(\boldsymbol{\Lambda})\mathbf{U}\mathbf{U}^{-1}\mathbf{f}(\boldsymbol{\Lambda})\mathbf{U} = \mathbf{g}(\mathbf{A})\mathbf{f}(\mathbf{A}).$$

Теорема 6. (Формула Кели — Гамильтона). Всякая матрица удовлетворяет своему характеристическому уравнению. Пусть $Q(\lambda) = (-1)^m \lambda^m + b_1\lambda^{m-1} + \ldots + b_m$ — характеристический полином, а $Q(\lambda) = 0$ — характеристическое уравнение. Теорема утверждает, что

$$Q(\mathbf{A}) = (-1)^m \mathbf{A}^m + b_1\mathbf{A}^{m-1} + \ldots + b_m\mathbf{E} \equiv \mathbf{0}.$$

Доказательство. Матрица простой структуры подобна диагональной матрице: $\mathbf{A} = \mathbf{U}\boldsymbol{\Lambda}\mathbf{U}^{-1}$. По теореме 2 о подобных матрицах $\mathbf{A}^k = \mathbf{U}\boldsymbol{\Lambda}^k\mathbf{U}^{-1}$. Подставим в

$$Q(\mathbf{A}) = (-1)^m \mathbf{A}^m + b_1\mathbf{A}^{m-1} + \ldots + b_m\mathbf{E}$$

вместо $\mathbf{A}$ ее выражение через $\mathbf{\Lambda}$:

$$Q(\mathbf{A}) = (-1)^m \mathbf{U}\mathbf{\Lambda}^m\mathbf{U}^{-1} + b_1\mathbf{U}\mathbf{\Lambda}^{m-1}\mathbf{U}^{-1} + \ldots + b_m\mathbf{E} =$$
$$= \mathbf{U}\Big((-1)^m \mathbf{\Lambda}^m + b_1\mathbf{\Lambda}^{m-1} + \ldots + b_m\mathbf{E}\Big)\mathbf{U}^{-1}.$$

В скобках стоит диагональная матрица с характеристическими полиномами на главной диагонали, в которые подставлены собственные значения, и, значит, тождественно равными нулю. Тогда матрица в скобках — нулевая, и теорема доказана.

Теорема 7. (Формула Лагранжа — Сильвестра). Любая функция матрицы $\mathbf{A}$, имеющей различные собственные значения, может быть представлена в виде:

$$\mathbf{f}(\mathbf{A}) = \sum_{k=1}^{m} f(\lambda_k)\frac{(\mathbf{A}-\lambda_1\mathbf{E})\ldots(\mathbf{A}-\lambda_{k-1}\mathbf{E})(\mathbf{A}-\lambda_{k+1}\mathbf{E})\ldots(\mathbf{A}-\lambda_m\mathbf{E})}{(\lambda_k-\lambda_1)\ldots(\lambda_k-\lambda_{k-1})(\lambda_k-\lambda_{k+1})\ldots(\lambda_k-\lambda_m)}. \quad (П3.10)$$

Доказательство. Представим функцию $f(x)$ в виде интерполяционного полинома Лагранжа $L_{m-1}(x)$, взяв в качестве узлов собственные значения матрицы $\mathbf{A}$: λ_1, λ_2, ..., λ_m

$$f(x) = L_{m-1}(x) + R_{m-1}(x).$$

Подставим в эту формулу $\mathbf{A}$ вместо x:

$$\mathbf{f}(\mathbf{A}) = \mathbf{L}_{m-1}(\mathbf{A}) + \mathbf{R}_{m-1}(\mathbf{A}).$$

Остаточный член $\mathbf{R}_{m-1}(\mathbf{A})$ принимает вид:

$$\mathbf{R}_{m-1}(\mathbf{A}) = \frac{f^{(m)}(\xi)}{m!}\omega(\mathbf{A}),$$

где $\omega(\mathbf{A}) = (\mathbf{A}-\lambda_1\mathbf{E})(\mathbf{A}-\lambda_2\mathbf{E})\ldots(\mathbf{A}-\lambda_{m-1}\mathbf{E})(\mathbf{A}-\lambda_m\mathbf{E})$. По теореме Кели — Гамильтона $\omega(\mathbf{A}) = \mathbf{0}$ и, следовательно, $\mathbf{f}(\mathbf{A}) = \mathbf{L}_{m-1}(\mathbf{A})$.

$$\mathbf{f}(\mathbf{A}) = \sum_{k=1}^{m} f(\lambda_k)\frac{(\mathbf{A}-\lambda_1\mathbf{E})\ldots(\mathbf{A}-\lambda_{k-1}\mathbf{E})(\mathbf{A}-\lambda_{k+1}\mathbf{E})\ldots(\mathbf{A}-\lambda_m\mathbf{E})}{(\lambda_k-\lambda_1)\ldots(\lambda_k-\lambda_{k-1})(\lambda_k-\lambda_{k+1})\ldots(\lambda_k-\lambda_m)} =$$
$$= \sum_{k=1}^{m} f(\lambda_k)\mathbf{S}_k(\mathbf{A},\lambda_k).$$

Теорема доказана. Отметим лишь, что матричные множители $\mathbf{S}_k(\mathbf{A},\lambda_k)$ для любых матричных функций остаются одинаковыми.

Формулу Лагранжа — Сильвестра можно модифицировать и на случай кратных собственных значений, осуществив предельный переход, при котором близкие собственные значения стремятся к общему значению. Приведем без доказательства формулы Лагранжа — Сильвестра для матриц второго и третьего порядка в случае кратных собственных значений.

Матрицы второго порядка. $\lambda_1 = \lambda_2$.

- У матрицы один элементарный делитель:

$$\mathbf{f}(\mathbf{A}) = f(\lambda_1)\mathbf{E} + \left(\mathbf{A} - \lambda_1\mathbf{E}\right) f'(\lambda_1). \qquad (\Pi 3.11)$$

- У матрицы два простых делителя:

$$\mathbf{f}(\mathbf{A}) = f(\lambda_1)\mathbf{E}.$$

Матрица третьего порядка. λ_1 — кратности 2, λ_2 — простое.

- λ_1 соответствует один элементарный делитель $\left(\lambda - \lambda_1\right)^2$:

$$\mathbf{f}(\mathbf{A}) = \frac{f(\lambda_1)\mathbf{E} + f'(\lambda_1)(\mathbf{A} - \lambda_1\mathbf{E})}{\lambda_1 - \lambda_2}(\mathbf{A} - \lambda_2\mathbf{E}) + \frac{f(\lambda_2)}{(\lambda_2 - \lambda_1)^2}(\mathbf{A} - \lambda_1\mathbf{E})^2.$$

- λ_1 соответствуют два элементарных делителя $\left(\lambda - \lambda_1\right)$:

$$\mathbf{f}(\mathbf{A}) = \frac{\mathbf{A} - \lambda_1\mathbf{E}}{\lambda_2 - \lambda_1} f(\lambda_2) + \frac{\mathbf{A} - \lambda_2\mathbf{E}}{\lambda_1 - \lambda_2} f(\lambda_1).$$

До сих пор мы ограничивались лишь такими матричными функциями, которые представляются сходящимися степенными рядами. Расширим класс рассматриваемых матричных функций.

Определение. Пусть матрица $\mathbf{A}$ не имеет кратных собственных значений, функция $f(\lambda)$ определена на спектре матрицы $\mathbf{A}$ (т. е. $f(\lambda)$ определена в точках λ_1, λ_2, ..., λ_m), а $L_{m-1}(x)$ — интерполяционный полином Лагранжа с узлами интерполирования λ_k. Тогда матричная функция $\mathbf{f}(\mathbf{A})$ равна $\mathbf{L}_{m-1}(\mathbf{A})$, т. е. имеет место формула (П3.10).

Как можно заметить на примере (П3.11), в случае кратных собственных значений потребуется, чтобы в точках λ_k была определена не только сама функция $f(\lambda)$, но и ее производные до известного порядка, определяемого кратностью λ_k.

В качестве упражнения проиллюстрируем некоторые из рассмотренных теорем на примере матрицы $\mathbf{A}=\begin{pmatrix}-51 & -49\\ -50 & -50\end{pmatrix}$.

Характеристический полином $\det(\mathbf{A}-\lambda\mathbf{E})=\lambda^2+101\lambda+100=0$, $\lambda_1=-100$, $\lambda_2=-1$.

Теорема 4. λ_1, λ_2.

- $\mathbf{A}: -100\ -1$.
- $\mathbf{e}^{\mathbf{A}}: e^{-100}e^{-1}$.
- $\mathbf{e}^{\mathbf{A}t}: e^{-100t}e^{-t}$.
- $\mathbf{A}^{-1}: (-100)^{-1}(-1)^{-1}$.
- $\mathbf{A}^n: (-100)^n(-1)^n$.
- $\mathbf{A}^T\mathbf{A}: (-100)(-100)(-1)(-1)$.
- $(\mathbf{E}-\mathbf{A})^{-1}: (1-(-100))^{-1}(1-(-1))^{-1}$.

Теорема 6. $\mathbf{A}^2+101\mathbf{A}+100\mathbf{E}=\mathbf{0}$. Воспользуемся этой теоремой для нахождения $\mathbf{A}^{-1}$. Умножим уравнение на $\mathbf{A}^{-1}$: $\mathbf{A}+101\mathbf{E}+100\mathbf{A}^{-1}=\mathbf{0}$ или

$$\mathbf{A}^{-1}=-(\mathbf{A}+101\mathbf{E})/100,$$

$$\mathbf{A}^{-1}=-\frac{1}{100}\left(\begin{pmatrix}-51 & -49\\ -50 & -50\end{pmatrix}+\begin{pmatrix}101 & 0\\ 0 & 101\end{pmatrix}\right)=-\frac{1}{100}\begin{pmatrix}50 & -49\\ -50 & 51\end{pmatrix}.$$

Теорема 7. Имеют место равенства:

$$\mathbf{f}(\mathbf{A})=f(\lambda_1)\frac{\mathbf{A}-\lambda_2\mathbf{E}}{\lambda_1-\lambda_2}+f(\lambda_2)\frac{\mathbf{A}-\lambda_1\mathbf{E}}{\lambda_2-\lambda_1},$$

$$\frac{\mathbf{A}-\lambda_2\mathbf{E}}{\lambda_1-\lambda_2}=-\frac{1}{99}\left(\begin{pmatrix}-51 & -49\\ -50 & -50\end{pmatrix}-\begin{pmatrix}-1 & 0\\ 0 & -1\end{pmatrix}\right)=\frac{1}{99}\begin{pmatrix}50 & 49\\ 50 & 49\end{pmatrix},$$

$$\frac{\mathbf{A}-\lambda_1\mathbf{E}}{\lambda_2-\lambda_1}=\frac{1}{99}\left(\begin{pmatrix}-51 & -49\\ -50 & -50\end{pmatrix}-\begin{pmatrix}-100 & 0\\ 0 & -100\end{pmatrix}\right)=\frac{1}{99}\begin{pmatrix}49 & -49\\ -50 & 50\end{pmatrix},$$

$$e^{\mathbf{A}t}=e^{-100t}\frac{1}{99}\begin{pmatrix}50 & 49\\ 50 & 49\end{pmatrix}+e^{-t}\frac{1}{99}\begin{pmatrix}49 & -49\\ -50 & 50\end{pmatrix}.$$

Следствие 2. Справедливо:

$$\|\mathbf{A}\|_1 = \max_j \sum_{i=1}^{m} |a_{ij}| = \max(101, 99) = 101;$$

$$\|\mathbf{A}\|_2 = (\rho(\mathbf{A}^* \mathbf{A}))^{1/2} = \left(\lambda_{\max}\left(\mathbf{A}^T \mathbf{A}\right)\right)^{1/2} = ((-100)(-100))^{1/2} = 100;$$

$$\|\mathbf{A}\|_\infty = \max_i \sum_{j=1}^{m} |a_{ij}| = \max(100, 100) = 100;$$

$$\|\mathbf{A}\|_E = \left(\sum_{i=1}^{m}\sum_{j=1}^{m} |a_{ij}|^2\right)^{1/2} = (51^2 + 2 \cdot 50 + 49^2)^{1/2} = (10002)^{1/2} > 100;$$

$$\|\mathbf{A}\|_M = m \cdot \max|a_{ij}| = 2 \cdot 51 = 102.$$

Как видно для всех норм, неравенство $\|\mathbf{A}\| \geq |\lambda|$ выполняется.

В дальнейшем мы будем широко пользоваться операциями дифференцирования и интегрирования матичных функций скалярного аргумента по этому аргументу. Такие операции выполняются поэлементно:

$$\mathbf{F}'(t) = \begin{pmatrix} F'_{11}(t) & F'_{12}(t) & . & . & F'_{1m}(t) \\ F'_{21}(t) & F'_{22}(t) & . & . & F'_{2m}(t) \\ . & . & . & . & . \\ . & . & . & . & . \\ F'_{m1}(t) & F'_{m2}(t) & . & . & F'_{mm}(t) \end{pmatrix};$$

$$\int \mathbf{F}(t)dt = \begin{pmatrix} \int F_{11}(t)dt & \int F_{12}(t)dt & . & . & \int F_{1m}(t)dt \\ \int F_{21}(t)dt & \int F_{22}(t)dt & . & . & \int F_{2m}(t)dt \\ . & . & . & . & . \\ . & . & . & . & . \\ \int F_{m1}(t)dt & \int F_{m2}(t)dt & . & . & \int F_{mm}(t)dt \end{pmatrix}.$$

В частности справедливы следующие формулы:

$$\frac{d}{dt}(\mathbf{F}(t) \cdot \mathbf{G}(t)) = \frac{d\mathbf{F}(t)}{dt}\mathbf{G}(t) + \mathbf{F}(t)\frac{d\mathbf{G}(t)}{dt};$$

$$\frac{d\mathbf{F}^k(t)}{dt} = \frac{d\mathbf{F}(t)}{dt}\mathbf{F}^{k-1}(t) + \mathbf{F}(t)\frac{d\mathbf{F}(t)}{dt}\mathbf{F}^{k-2}(t) + \ldots + \mathbf{F}^{k-1}(t)\frac{d\mathbf{F}(t)}{dt};$$

$$\frac{d\mathbf{F}^{-1}(t)}{dt} = -\mathbf{F}^{-1}(t)\frac{d\mathbf{F}(t)}{dt}\mathbf{F}^{-1}(t).$$

Первые две формулы очевидны, третью рекомендуется доказать самостоятельно.

П3.6. Некоторые свойства матричной экспоненты

Матричная экспонента является одной из наиболее употребительных матричных функций, поэтому обратимся к некоторым ее свойствам. Их справедливость может быть установлена проведением ряда операций с матричными рядами. Опуская строгие доказательства, ограничимся лишь указанием путей их построения.

1. $e^{\mathbf{A}} \cdot e^{\mathbf{B}} \neq e^{\mathbf{A}+\mathbf{B}}$.

   Доказательство. Выпишем матричные разложения $e^{\mathbf{A}}$ и $e^{\mathbf{B}}$ и перемножим их, ограничившись членами вплоть до квадратичных.

$$e^{\mathbf{A}} \cdot e^{\mathbf{B}} = \left(\mathbf{E}+\mathbf{A}+\frac{\mathbf{A}^2}{2!}+\ldots\right)\cdot\left(\mathbf{E}+\mathbf{B}+\frac{\mathbf{B}^2}{2!}+\ldots\right)=$$

$$=\mathbf{E}+\mathbf{A}+\frac{\mathbf{A}^2}{2!}+\ldots+\mathbf{B}+\mathbf{AB}+\frac{\mathbf{A}^2\mathbf{B}}{2!}+\ldots+\frac{\mathbf{B}^2}{2!}+\frac{\mathbf{AB}^2}{2!}+\ldots$$

   Теперь запишем матричный ряд для $e^{\mathbf{A}+\mathbf{B}}$:

$$e^{\mathbf{A}+\mathbf{B}} = \mathbf{E}+\mathbf{A}+\mathbf{B}+\frac{(\mathbf{A}+\mathbf{B})^2}{2!}+\ldots$$

   и сравним ряды. В произведении $e^{\mathbf{A}} \cdot e^{\mathbf{B}}$ квадратичные слагаемые равны $\frac{\mathbf{A}^2}{2!}+\mathbf{AB}+\frac{\mathbf{B}^2}{2!}$, а в разложении $e^{\mathbf{A}+\mathbf{B}}$ они имеют вид $\frac{(\mathbf{A}+\mathbf{B})^2}{2!}=\frac{\mathbf{A}^2}{2!}+\frac{\mathbf{AB}+\mathbf{BA}}{2!}+\frac{\mathbf{B}^2}{2!}$. Совпадение будет, только если $\mathbf{AB}=\mathbf{BA}$.

   В общем случае, когда матрицы не коммутируют $(\mathbf{AB}\neq\mathbf{BA})$, $e^{\mathbf{A}} \cdot e^{\mathbf{B}} \neq e^{\mathbf{A}+\mathbf{B}}$.

2. $e^{\mathbf{A}t} \cdot e^{\mathbf{A}q} = e^{\mathbf{A}(t+q)}$. Здесь t, q — скалярные величины.

Доказательство. Воспользуемся предыдущими результатами.

$$e^{\mathbf{A}t} \cdot e^{\mathbf{A}q} = \mathbf{E} + \mathbf{A}t + \mathbf{A}q + \frac{\mathbf{A}^2 t^2}{2!} + \mathbf{A}t\mathbf{A}q + \frac{\mathbf{A}^2 q^2}{2!} + \ldots =$$

$$= \mathbf{E} + \mathbf{A}(t+q) + \frac{\mathbf{A}^2 (t+q)^2}{2!} + \ldots$$

$$e^{\mathbf{A}(t+q)} = \mathbf{E} + \mathbf{A}t + \mathbf{A}q + \frac{\mathbf{A}^2 (t+q)^2}{2!} + \ldots = \mathbf{E} + \mathbf{A}(t+q) + \frac{\mathbf{A}^2 (t+q)^2}{2!} + \ldots$$

Здесь проблемы с коммутативностью матриц нет, и результат очевиден.

3. $\left(e^{\mathbf{A}}\right)^{-1} = e^{-\mathbf{A}}$.

Доказательство этого свойства состоит в проверке равенства $e^{\mathbf{A}} \cdot e^{-\mathbf{A}} = \mathbf{E}$, т. е. в непосредственном перемножении рядов для обеих матриц. Рекомендуется выполнить эти преобразования самостоятельно.

4. $\frac{d}{dt} e^{\mathbf{A}t} = \mathbf{A} e^{\mathbf{A}t} = e^{\mathbf{A}t} \mathbf{A}$.

Доказательство.

$$\frac{d}{dt} e^{\mathbf{A}t} = \frac{d}{dt}\left(\mathbf{E} + \mathbf{A}t + \frac{\mathbf{A}^2 t^2}{2!} + \frac{\mathbf{A}^3 t^3}{3!} + \ldots \right) = \mathbf{A} + \mathbf{A}^2 t + \frac{\mathbf{A}^3 t^2}{2!} + \ldots =$$

$$= \mathbf{A}\left(\mathbf{E} + \mathbf{A}t + \frac{\mathbf{A}^2 t^2}{2!} + \frac{\mathbf{A}^3 t^3}{3!} + \ldots \right) =$$

$$= \mathbf{A} e^{\mathbf{A}t} = \left(\mathbf{E} + \mathbf{A}t + \frac{\mathbf{A}^2 t^2}{2!} + \frac{\mathbf{A}^3 t^3}{3!} + \ldots \right)\mathbf{A} = e^{\mathbf{A}t}\mathbf{A}.$$

5. $\int_0^T e^{\mathbf{A}t} dt = \mathbf{A}^{-1}\left(e^{\mathbf{A}T} - \mathbf{E}\right) = \left(e^{\mathbf{A}T} - \mathbf{E}\right)\mathbf{A}^{-1}$.

Доказательство.

$$\int_0^T e^{\mathbf{A}t} dt = \int_0^T \left(\mathbf{E} + \mathbf{A}t + \frac{\mathbf{A}^2 t^2}{2!} + \frac{\mathbf{A}^3 t^3}{3!} + \ldots \right) dt = \left(\mathbf{E}T + \frac{\mathbf{A}T^2}{2!} + \frac{\mathbf{A}^2 T^3}{3!} + \ldots \right).$$

Умножим обе части равенства на $\mathbf{A}$ слева, а затем в правой части прибавим и вычтем $\mathbf{E}$.

$$\mathbf{A}\int_0^T e^{\mathbf{A}t}dt = \mathbf{A}T + \frac{\mathbf{A}^2T^2}{2!} + \frac{\mathbf{A}^3T^3}{3!} + \ldots + \mathbf{E} - \mathbf{E} \Rightarrow \int_0^T e^{\mathbf{A}t}dt = \mathbf{A}^{-1}\left(e^{\mathbf{A}T} - \mathbf{E}\right).$$

Умножение на $\mathbf{A}$ можно выполнить и справа, и тогда:

$$\int_0^T e^{\mathbf{A}t}dt = \left(e^{\mathbf{A}T} - \mathbf{E}\right)\mathbf{A}^{-1}.$$

Выражение справа требует существования обратной матрицы $\mathbf{A}$. Однако интеграл существует для любой матрицы. Если $\det(\mathbf{A}) = 0$, можно использовать другой способ записи результата, ограничиваясь непосредственным интегрированием ряда без введения $\mathbf{A}^{-1}$.

П3.7. Аналитическое решение систем линейных дифференциальных уравнений с постоянной матрицей

Рассмотрим систему обыкновенных дифференциальных уравнений первого порядка, разрешенную относительно производных

$$\frac{dx^{(i)}(t)}{dt} = f^{(i)}(t, x^{(1)}(t), x^{(2)}(t), \ldots, x^{(m)}(t)), \quad i = 1, 2, \ldots, m,$$

где t — независимая переменная, $x^{(1)}(t)$, $x^{(2)}(t)$, ..., $x^{(m)}(t)$ — искомые функции, $f^{(i)}$ — функции, определенные на некотором открытом множестве $\mathbf{G}$ ($m+1$)-мерного евклидова пространства переменных t, $x^{(1)}(t)$, $x^{(2)}(t)$, ..., $x^{(m)}(t)$. Номер компонента вектора здесь везде будем писать как верхний индекс в скобках. Перейдя к векторно-матричным обозначениям

$$\mathbf{x}(t) = \begin{pmatrix} x^{(1)}(t) \\ x^{(2)}(t) \\ \ldots \\ x^{(m)}(t) \end{pmatrix}, \qquad \mathbf{f}(t,\mathbf{x}) = \begin{pmatrix} f^{(1)}(t,x) \\ f^{(2)}(t,x) \\ \ldots \\ f^{(m)}(t,x) \end{pmatrix},$$

исходную систему перепишем в виде

$$\frac{d\mathbf{x}(t)}{dt} = \mathbf{f}(t, \mathbf{x}). \qquad (П3.12)$$

При этом требуется найти решение $\mathbf{x}(t)$, удовлетворяющее начальным условиям $\mathbf{x}(t_0) = \mathbf{x}_0$. Такая задача называется *начальной задачей* или *задачей Коши*.

Важным классом дифференциальных систем являются линейные системы:

$$\frac{dx^{(i)}(t)}{dt} = a_{i1}(t)x^{(1)}(t) + a_{i2}(t)x^{(2)}(t) + \ldots + a_{im}(t)x^{(m)}(t) + g^{(i)}(t), \; i = 1, 2, \ldots, m,$$

где $a_{ij}(t)$ и $g^{(i)}(t)$ — непрерывные функции t. Введя дополнительные обозначения матрицы $\mathbf{A}(t)$ с элементами $a_{ij}(t)$, а также вектора $\mathbf{g}(t) = \left(g^{(1)}(t), g^{(2)}(t), \ldots, g^{(m)}(t)\right)^T$, получим уравнения в векторно-матричном виде:

$$\frac{d\mathbf{x}(t)}{dt} = \mathbf{A}(t)\mathbf{x}(t) + \mathbf{g}(t), \qquad \mathbf{x}(t_0) = \mathbf{x}_0 .$$

Если $a_{ij}(t) = \text{const}$, система называется линейной дифференциальной системой с постоянной матрицей или постоянными коэффициентами:

$$\frac{d\mathbf{x}(t)}{dt} = \mathbf{A}\mathbf{x}(t) + \mathbf{g}(t), \qquad \mathbf{x}(t_0) = \mathbf{x}_0. \qquad (П3.13)$$

Сначала обратимся к однородной системе

$$\frac{d\mathbf{x}(t)}{dt} = \mathbf{A}\mathbf{x}(t) .$$

Ее решением является функция $\mathbf{x}(t) = e^{\mathbf{A}t}\mathbf{c}$, где $\mathbf{c}$ — вектор произвольных постоянных. Убедиться в этом можно непосредственной подстановкой решения в уравнение. Выражение слева $\frac{d}{dt}\left(e^{\mathbf{A}t}\mathbf{c}\right) = \mathbf{A}e^{\mathbf{A}t}\mathbf{c}$ равно выражению справа $\mathbf{A}e^{\mathbf{A}t}\mathbf{c}$.

Неоднородная система решается методом Лагранжа вариации произвольных постоянных. При этом полагаем, что элементы вектора $\mathbf{c}$ являются функция-

ми независимой переменной $\mathbf{c} = \mathbf{c}(t)$. Подставляем искомый вид решения в уравнение

$$\mathbf{A}e^{\mathbf{A}t}\mathbf{c}(t) + e^{\mathbf{A}t}\frac{d\mathbf{c}(t)}{dt} = \mathbf{A}e^{\mathbf{A}t}\mathbf{c}(t) + \mathbf{g}(t).$$

Отсюда $e^{\mathbf{A}t}\frac{d\mathbf{c}(t)}{dt} = \mathbf{g}(t)$, и после умножения обеих частей равенства на $e^{-\mathbf{A}t}$ получаем:

$$\frac{d\mathbf{c}(t)}{dt} = e^{-\mathbf{A}t}\mathbf{g}(t).$$

Интегрируем это уравнение от t_0 до t:

$$\mathbf{c}(t) - \mathbf{c}(t_0) = \int_{t_0}^{t} e^{-\mathbf{A}\tau}\mathbf{g}(\tau)d\tau,$$

и, подставив $\mathbf{c}(t)$ в искомый вид решения, определяем общее решение линейной неоднородной дифференциальной системы:

$$\mathbf{x}(t) = e^{\mathbf{A}t}\mathbf{c}(t_0) + e^{\mathbf{A}t}\int_{t_0}^{t} e^{-\mathbf{A}\tau}\mathbf{g}(\tau)d\tau.$$

Учитывая начальные условия, находим вектор $\mathbf{c}(t_0)$: $\mathbf{x}(t_0) = \mathbf{x}_0 = e^{\mathbf{A}t_0}\mathbf{c}(t_0)$ или $\mathbf{c}(t_0) = e^{-\mathbf{A}t_0}\mathbf{x}_0$ и окончательно получаем:

$$\mathbf{x}(t) = e^{\mathbf{A}(t-t_0)}\mathbf{x}_0 + \int_{t_0}^{t} e^{\mathbf{A}(t-\tau)}\mathbf{g}(\tau)d\tau.$$

Без нарушения общности можно считать, что начальным значением независимой переменной является $t_0 = 0$. Тогда

$$\mathbf{x}(t) = e^{\mathbf{A}t}\mathbf{x}_0 + \int_{0}^{t} e^{\mathbf{A}(t-\tau)}\mathbf{g}(\tau)d\tau = e^{\mathbf{A}t}\mathbf{x}_0 + \int_{0}^{t} e^{\mathbf{A}\tau}\mathbf{g}(t-\tau)d\tau. \qquad (\text{П}3.14)$$

Последнее преобразование использует теорему о свертке:

$$\int_{0}^{t} f(t-\tau)g(\tau)d\tau = \int_{0}^{t} f(\tau)g(t-\tau)d\tau,$$

в справедливости которой легко убедиться простой заменой переменной: $t-\tau=\tau$.

Считая вектор $\mathbf{g}(t)$ постоянным $\left(\mathbf{g}(t)=\mathbf{g}=\text{const}\right)$, упростим равенство (П3.14):

$$\mathbf{x}(t)=e^{\mathbf{A}t}\mathbf{x}_0+\int_0^t e^{\mathbf{A}(t-\tau)}d\tau\cdot\mathbf{g}=e^{\mathbf{A}t}\mathbf{x}_0+\left(e^{\mathbf{A}t}-\mathbf{E}\right)\mathbf{A}^{-1}\mathbf{g}.$$

Теперь запишем полученное решение в ином виде, воспользовавшись биортогональной системой собственных векторов матрицы $\mathbf{A}$. Как и в *разд. П3.3*, правые собственные векторы будем обозначать $\mathbf{u}_i$, а левые — $\mathbf{v}_i$, $i=1,\ 2,\ldots,\ m$. При этом считаем, что векторы ортонормированы так, что $\mathbf{u}_i^T\mathbf{v}_i=1$. Запишем разложения векторов $\mathbf{x}(t)$, $\mathbf{g}(t)$ и $\mathbf{x}_0$ по биортогональной системе векторов матрицы $\mathbf{A}$:

$$\mathbf{x}(t)=\sum_{i=1}^{m}\left(\mathbf{x}^T\mathbf{v}_i\right)\cdot\mathbf{u}_i;\quad \mathbf{g}(t)=\sum_{i=1}^{m}\left(\mathbf{g}^T\mathbf{v}_i\right)\cdot\mathbf{u}_i;$$

$$\mathbf{x}_0(t)=\sum_{i=1}^{m}\left(\mathbf{x}_0^T\mathbf{v}_i\right)\cdot\mathbf{u}_i;,\quad \frac{d\mathbf{x}(t)}{dt}=\sum_{i=1}^{m}\left(\frac{d\mathbf{x}(t)^T}{dt}\mathbf{v}_i\right)\cdot\mathbf{u}_i$$

и подставим их в исходное уравнение (П3.13):

$$\sum_{i=1}^{m}\left(\frac{d\mathbf{x}(t)^T}{dt}\mathbf{v}_i\right)\cdot\mathbf{u}_i=\sum_{i=1}^{m}\mathbf{A}\left(\mathbf{x}^T\mathbf{v}_i\right)\cdot\mathbf{u}_i+\sum_{i=1}^{m}\left(\mathbf{g}^T\mathbf{v}_i\right)\cdot\mathbf{u}_i.$$

Вынесем операцию дифференцирования за скобки и учтем, что $\mathbf{A}\mathbf{u}_i=\lambda_i\mathbf{u}_i$:

$$\sum_{i=1}^{m}\frac{d}{dt}\left(\mathbf{x}^T(t)\mathbf{v}_i\right)\cdot\mathbf{u}_i=\sum_{i=1}^{m}\lambda_i\left(\mathbf{x}^T\mathbf{v}_i\right)\cdot\mathbf{u}_i+\sum_{i=1}^{m}\left(\mathbf{g}^T\mathbf{v}_i\right)\cdot\mathbf{u}_i.$$

Выпишем отдельно уравнение для i-ого компонента:

$$\frac{d}{dt}\left(\mathbf{x}^T\mathbf{v}_i\right)=\lambda_i\left(\mathbf{x}^T\mathbf{v}_i\right)+\left(\mathbf{g}^T\mathbf{v}_i\right).$$

Это линейное дифференциальное уравнение первого порядка относительно скалярной функции $\left(\mathbf{x}^T\mathbf{v}_i\right)$. Его решение с учетом начальных условий $\left.\left(\mathbf{x}^T\mathbf{v}_i\right)\right|_{t=0}=\left(\mathbf{x}_0^T\mathbf{v}_i\right)$ имеет вид:

$$\left(\mathbf{x}^T(t)\mathbf{v}_i\right)=e^{\lambda_i t}\left(\mathbf{x}_0^T\mathbf{v}_i\right)+\int_0^t e^{\lambda_i(t-\tau)}\left(\mathbf{g}(\tau)^T\mathbf{v}_i\right)d\tau.$$

Окончательно для $\mathbf{x}(t)$ получаем

$$\mathbf{x}(t)=\sum_{i=1}^{m}\left(\mathbf{x}^T\mathbf{v}_i\right)\mathbf{u}_i=\sum_{i=1}^{m}\left(e^{\lambda_i t}\left(\mathbf{x}_0^T\mathbf{v}_i\right)+\int_0^t e^{\lambda_i(t-\tau)}\left(\mathbf{g}(\tau)^T\mathbf{v}_i\right)d\tau\right)\cdot\mathbf{u}_i. \quad (\text{П3.15})$$

Сравнивая два решения (П3.14) и (П3.15) и учитывая формулу (П3.10), можно связать матричные множители в формуле Лагранжа — Сильвестра и собственные векторы $\mathbf{u}_i$ и $\mathbf{v}_i$. Выпишем первые слагаемые в обеих формулах, определяющие решение однородной системы:

$$\mathbf{x}(t)=e^{\mathbf{A}t}\mathbf{x}_0=\sum_{i=1}^{m}e^{\lambda_i t}S_i(\mathbf{A},\lambda_i)\cdot\mathbf{x}_0,$$

$$\mathbf{x}(t)=\sum_{i=1}^{m}e^{\lambda_i t}\mathbf{u}_i(\mathbf{x}_0{}^T\mathbf{v}_i)=\sum_{i=1}^{m}e^{\lambda_i t}\mathbf{u}_i(\mathbf{v}_i^T\mathbf{x}_0). \quad (\text{П3.16})$$

Отсюда $\mathbf{S}_i\left(\mathbf{A},\lambda_i\right)=\mathbf{u}_i\cdot\mathbf{v}_i^T$.

П3.8. Аналитическое решение систем линейных разностных уравнений с постоянной матрицей

Рассмотрим линейную систему разностных уравнений с переменными коэффициентами или переменной матрицей

$$y^{(i)}(k+1)=b_{i1}(k)y^{(1)}(k)+b_{i2}(k)y^{(2)}(k)+\ldots+b_{im}(k)y^{(m)}(k)+g^{(i)}(k),$$
$$i=1,\ 2,\ \ldots,\ m,$$

где k — независимая целочисленная переменная, $y^{(i)}(k)$ — искомые функции, $b_{ij}(k)$ и $g^{(i)}(k)$ — заданные функции целочисленного аргумента.

Введя дополнительные обозначения матрицы $\mathbf{B}(k)$ с элементами $b_{ij}(k)$, а также векторов $\mathbf{y}(k)$ и $\mathbf{g}(k)$

$$\mathbf{y}(k)=\left(y^{(1)}(k),y^{(2)}(k),\ldots,y^{(m)}(k)\right)^T,\quad \mathbf{g}(k)=\left(g^{(1)}(k),g^{(2)}(k),\ldots,g^{(m)}(k)\right)^T,$$

перепишем систему в виде

$$\mathbf{y}(k+1)=\mathbf{B}(k)\mathbf{y}(k)+\mathbf{g}(k). \qquad (П3.17)$$

Функцию $\mathbf{y}(k)$, удовлетворяющую при $k=0, 1, 2, ..., n, ...$ уравнению (П3.17), называют *решением линейной разностной системы*. Если заданы значения функции $\mathbf{y}(k)$ в некоторой точке k_0: $\mathbf{y}(k_0)=\left(y^{(1)}(k_0),\ y^{(2)}(k_0),\ ...,\ y^{(m)}(k_0)\right)^T$, то говорят, что относительно разностной системы поставлена начальная задача (аналог задачи Коши для дифференциальной системы). Начальные условия из бесконечного множества решений разностной системы выделяют одно, проходящее через эту точку.

Если коэффициенты матрицы постоянны: $b_{ij}(k)=\text{const}=b_{ij}$, то говорят о разностной системе с постоянной матрицей. Приступим к решению неоднородной разностной системы, относительно которой поставлена начальная задача: $\mathbf{y}(0)=\mathbf{y}_0$. Начинаем с однородной системы

$$\mathbf{y}(k+1)=\mathbf{B}\mathbf{y}(k).$$

Ее решением является функция $\mathbf{y}(k)=\mathbf{B}^k\mathbf{c}$, где $\mathbf{c}$ — вектор произвольных постоянных, что непосредственно проверяется подстановкой предложенного решения в уравнение: выражение слева $\mathbf{B}^{k+1}\mathbf{c}$ равно выражению справа $\mathbf{B}\mathbf{B}^k\mathbf{c}$.

Неоднородную систему

$$\mathbf{y}(k+1)=\mathbf{B}\mathbf{y}(k)+\mathbf{g}(k) \qquad (П3.18)$$

решаем методом вариации произвольных постоянных, т. е. полагаем $\mathbf{c}=\mathbf{c}(k)$. Искомый вид решения $\mathbf{y}(k)=\mathbf{B}^k\mathbf{c}(k)$ подставляем в уравнение

$$\mathbf{B}^{k+1}\mathbf{c}(k+1)=\mathbf{B}\mathbf{B}^k\mathbf{c}(k)+\mathbf{g}(k).$$

Группируя слагаемые, получаем $\mathbf{B}^{k+1}\Delta\mathbf{c}(k)=\mathbf{g}(k)$ или $\Delta\mathbf{c}(k)=\mathbf{B}^{-k-1}\mathbf{g}(k)$. Суммируем последнее равенство

$$\sum_{i=0}^{k-1}\Delta\mathbf{c}(i)=\mathbf{c}(k)-\mathbf{c}(0)=\sum_{i=0}^{k-1}\mathbf{B}^{-i-1}\mathbf{g}(i)$$

и для решения $\mathbf{y}(k)$ имеем

$$\mathbf{y}(k)=\mathbf{B}^k\mathbf{c}(0)+\mathbf{B}^k\sum_{i=0}^{k-1}\mathbf{B}^{-i-1}\mathbf{g}(i).$$

Для нахождения вектора произвольных постоянных $\mathbf{c}(0)$ воспользуемся начальными условиями и получим $\mathbf{y}(0) = \mathbf{y}_0 = \mathbf{c}(0)$ и

$$\mathbf{y}(k) = \mathbf{B}^k \mathbf{y}(0) + \sum_{i=0}^{k-1} \mathbf{B}^{k-i-1}\mathbf{g}(i) = \mathbf{B}^k \mathbf{y}(0) + \sum_{i=0}^{k-1} \mathbf{B}^i \mathbf{g}(k-i-1). \qquad (\text{П}3.19)$$

В последнем равенстве изменением порядка суммирования получается результат, аналогичный тому, что дает теорема о свертке в непрерывном случае:

$$\sum_{i=0}^{k-1} \mathbf{B}^{k-i-1}\mathbf{g}(i) = \mathbf{B}^{k-1}\mathbf{g}(0) + \mathbf{B}^{k-2}\mathbf{g}(1) + \ldots + \mathbf{B}^1\mathbf{g}(k-2) + \mathbf{B}^0\mathbf{g}(k-1) =$$

$$= \mathbf{B}^0\mathbf{g}(k-1) + \mathbf{B}^1\mathbf{g}(k-2) + \ldots + \mathbf{B}^{k-2}\mathbf{g}(1) + \mathbf{B}^{k-1}\mathbf{g}(0) = \sum_{i=0}^{k-1} \mathbf{B}^i \mathbf{g}(k-i-1).$$

В частном случае, когда $\mathbf{g}(k) = \text{const} = \mathbf{g}$, этот вектор можно вынести за знак суммы:

$$\mathbf{y}(k) = \mathbf{B}^k \mathbf{y}(0) + \left(\sum_{i=0}^{k-1} \mathbf{B}^i\right)\mathbf{g} = \mathbf{B}^k \mathbf{y}(0) + \left(\mathbf{B}^k - \mathbf{E}\right)\left(\mathbf{B} - \mathbf{E}\right)^{-1}\mathbf{g}. \qquad (\text{П}3.20)$$

Последнее выражение было записано с учетом очевидного равенства.

$$\left(\mathbf{B} - \mathbf{E}\right)\left(\sum_{i=0}^{k-1} \mathbf{B}^i\right) = \left(\mathbf{B}^k - \mathbf{E}\right).$$

Как и в предыдущем разделе, решение может быть записано с использованием биортогональной системы собственных векторов. По-прежнему считаем, что выполнено нормирование, т. е. $\mathbf{u}_i^T \mathbf{v}_i = 1$. Выписываем разложения векторов $\mathbf{y}(k)$, $\mathbf{g}(k)$ и $\mathbf{y}_0$ по биортогональной системе векторов матрицы $\mathbf{B}$:

$$\mathbf{y}(k) = \sum_{i=1}^{m} \left(\mathbf{y}^T(k)\mathbf{v}_i\right) \cdot \mathbf{u}_i\,; \quad \mathbf{g}(k) = \sum_{i=1}^{m} \left(\mathbf{g}^T(k)\mathbf{v}_i\right) \cdot \mathbf{u}_i\,; \quad \mathbf{y}_0 = \sum_{i=1}^{m} \left(\mathbf{y}_0^T \mathbf{v}_i\right) \cdot \mathbf{u}_i$$

и подставляем их в уравнение (П3.18):

$$\sum_{i=1}^{m} \left(\mathbf{y}^T(k+1)\mathbf{v}_i\right) \cdot \mathbf{u}_i = \sum_{i=1}^{m} \mathbf{B}\left(\mathbf{y}^T(k)\mathbf{v}_i\right) \cdot \mathbf{u}_i + \sum_{i=1}^{m} \left(\mathbf{g}^T(k)\mathbf{v}_i\right) \cdot \mathbf{u}_i\,.$$

Учтем, что $\mathbf{B}\mathbf{u}_i = \lambda_i \mathbf{u}_i$ и выпишем уравнение для i-ого компонента:

$$\left(\mathbf{y}^T(k+1)\mathbf{v}_i\right) = \lambda_i \left(\mathbf{y}^T(k)\mathbf{v}_i\right) + \left(\mathbf{g}^T(k)\mathbf{v}_i\right).$$

Это линейное разностное уравнение первого порядка относительно скалярной функции $\left(\mathbf{y}^T(k)\mathbf{v}_i\right)$.

С учетом начальных условий $\left(\mathbf{y}^T(k)\mathbf{v}_i\right)\Big|_{k=0} = \left(\mathbf{y}_0^T\mathbf{v}_i\right)$ его решение имеет вид:

$$\left(\mathbf{y}^T(k)\mathbf{v}_i\right) = \lambda_i^k\left(\mathbf{y}_0^T\mathbf{v}_i\right) + \lambda_i^k\sum_{p=0}^{k-1}\lambda_i^{-p-1}\left(\mathbf{g}^T(p)\mathbf{v}_i\right).$$

И, наконец, получаем выражение для $\mathbf{y}(k)$:

$$\mathbf{y}(k) = \sum_{i=1}^{m}\left(\mathbf{y}^T(k)\mathbf{v}_i\right)\cdot\mathbf{u}_i = \sum_{i=1}^{m}\left(\lambda_i^k\left(\mathbf{y}_0^T\mathbf{v}_i\right) + \sum_{p=0}^{k-1}\lambda_i^{k-p-1}\left(\mathbf{g}^T(p)\mathbf{v}_i\right)\right)\cdot\mathbf{u}_i.$$

Как и в предыдущем разделе, здесь можно воспользоваться теоремой о свертке:

$$\mathbf{y}(k) = \sum_{i=1}^{m}\left(\lambda_i^k\left(\mathbf{y}_0^T\mathbf{v}_i\right) + \sum_{p=0}^{k-1}\lambda_i^{p}\left(\mathbf{g}^T(k-p-1)\mathbf{v}_i\right)\right)\cdot\mathbf{u}_i.$$

Таким образом, решение системы линейных дифференциальных и разностных уравнений может быть представлено как через матричные функции, так и посредством биортогональной системы собственных векторов.

В завершение раздела проиллюстрируем приведение линейных дифференциальных и разностных уравнений порядка выше первого к системе уравнений (табл. П3.1).

Таблица П3.1. *Приведение линейных дифференциальных и разностных уравнений порядка выше первого к системе уравнений*

Дифференциальное уравнение	**Разностное уравнение**
$z^{(s)}(t) = a_1 z^{(s-1)}(t) + a_2 z^{(s-2)}(t) + \ldots + a_s z(t) + g(t)$	$y(k+s) = b_1 y(k+s-1) + b_2 y(k+s-2) + \ldots + b_s y(k) + g(k)$
Начальные условия	
$z(0),\ z'(0),\ \ldots,\ z^{(s-1)}(0)$	$y(0),\ y(1),\ \ldots,\ y(s-1)$
Выполняем замену искомых функций	
$z(t) = \zeta_1(t)$	$y(k) = \upsilon_1(k)$
$z'(t) = \zeta_2(t)$	$y(k+1) = \upsilon_2(k)$
...	...
$z^{(s-1)}(t) = \zeta_s(t)$	$y(k+s-1) = \upsilon_s(k)$

Таблица П3.1 (окончание)

Дифференциальное уравнение	Разностное уравнение
Тогда	
$\zeta_1'(t) = z'(t) = \zeta_2(t)$	$\upsilon_1(k+1) = y(k+1) = \upsilon_2(k)$
$\zeta_2'(t) = z''(t) = \zeta_3(t)$	$\upsilon_2(k+1) = y(k+2) = \upsilon_3(k)$
...	...
$\zeta_s'(t) = z^{(s)}(t) = a_s\zeta_1(t) + a_{s-1}\zeta_2(t) + \ldots + a_1\zeta_s(t) + g(t)$	$\upsilon_s(k+1) = y(k+s) = b_s\upsilon_1(k) + b_{s-1}\upsilon_2(k) + \ldots + b_1\upsilon_s(k) + g(k)$

Осталось лишь использовать матрично-векторные обозначения и уравнения переписать в таком виде (табл. П3.2).

Таблица П3.2. *Использование матрично-векторных обозначений*

Дифференциальное уравнение	Разностное уравнение
$\dfrac{d\boldsymbol{\varsigma}(t)}{dt} = \mathbf{A}\boldsymbol{\varsigma}(t) + \mathbf{g}(t)$, $\boldsymbol{\varsigma}(0)$, где $\boldsymbol{\varsigma}(t) = \begin{pmatrix} \varsigma_1(t) \\ \varsigma_2(t) \\ \ldots \\ \ldots \\ \varsigma_s(t) \end{pmatrix}$; $\mathbf{g}(t) = \begin{pmatrix} 0 \\ 0 \\ \ldots \\ \ldots \\ g(t) \end{pmatrix}$; $\mathbf{A} = \begin{pmatrix} 0 & 1 & 0 & \ldots & 0 \\ 0 & 0 & 1 & \ldots & 0 \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ 0 & 0 & 0 & \ldots & 1 \\ a_s & a_{s-1} & a_{s-2} & \ldots & a_1 \end{pmatrix}$.	$\boldsymbol{\upsilon}(k+1) = \mathbf{B}\boldsymbol{\upsilon}(k) + \mathbf{g}(k)$, $\boldsymbol{\upsilon}(0)$, где $\boldsymbol{\upsilon}(k) = \begin{pmatrix} \upsilon_1(k) \\ \upsilon_2(k) \\ \ldots \\ \ldots \\ \upsilon_s(k) \end{pmatrix}$; $\mathbf{g}(k) = \begin{pmatrix} 0 \\ 0 \\ \ldots \\ \ldots \\ g(k) \end{pmatrix}$; $\mathbf{B} = \begin{pmatrix} 0 & 1 & 0 & \ldots & 0 \\ 0 & 0 & 1 & \ldots & 0 \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ 0 & 0 & 0 & \ldots & 1 \\ b_s & b_{s-1} & b_{s-2} & \ldots & b_1 \end{pmatrix}$.

Нумерацию новых функций при замене можно произвести и в другом порядке, начиная с $z^{(s-1)}(t)$ и $y(k+s-1)$. Тогда матрицы $\mathbf{A}$ и $\mathbf{B}$ и векторы $\varsigma(t)$, $\mathbf{v}(k)$, $\mathbf{g}(t)$ и $\mathbf{g}(k)$ будут иметь несколько иной вид.

П3.9. Устойчивость решений дифференциальных и разностных уравнений

Обратимся к системе нелинейных дифференциальных уравнений

$$\frac{d\mathbf{x}(t)}{dt} = \mathbf{f}(t,\mathbf{x}), \quad \mathbf{x}(t_0) = \mathbf{x}_0, \quad t \in [a,b], \tag{П3.21}$$

где t — независимая переменная, $\mathbf{x}$ — вектор решения; $\mathbf{f}(t,\mathbf{x})$ — вектор-функция, непрерывная по t и имеющая непрерывные частные производные первого порядка по компонентам вектора $\mathbf{x}$.

Большой интерес представляет исследование зависимости решения задачи Коши от начальных условий. Если незначительные изменения в $\mathbf{x}_0$ могут существенно изменить решение, то в прикладном отношении такое решение часто неприемлемо. На конечном промежутке $[a,\ b]$ для систем (П3.21) с непрерывной функцией $\mathbf{f}(t,\mathbf{x})$ и свойством единственности решения имеет место *интегральная непрерывность решений*. Иными словами, для любого $\varepsilon > 0$ существует такое $\delta > 0$, что для любых двух решений $\mathbf{x}(t)$ и $\mathbf{z}(t)$ системы (П3.21), отличающихся начальными условиями не более, чем на δ $\left(\|\mathbf{x}(t_0) - \mathbf{z}(t_0)\| < \delta\right)$, будет иметь место $\left(\|\mathbf{x}(t) - \mathbf{z}(t)\| < \varepsilon\right)$. Иначе обстоит дело для бесконечного промежутка при $t \to \infty$. Изучением этих вопросов занимается теория устойчивости.

Определение 1. Решение $\mathbf{x}(t)$ системы (П3.21) называется *устойчивым по Ляпунову*, если для любого $\varepsilon > 0$ найдется такое $\delta > 0$, что для всех решений $\mathbf{z}(t)$ системы (П3.21), удовлетворяющих неравенству $\left(\|\mathbf{x}(t_0) - \mathbf{z}(t_0)\| < \delta\right)$, справедливо неравенство $\left(\|\mathbf{x}(t) - \mathbf{z}(t)\| < \varepsilon\right)$ при всех $t \in [t_0, \infty)$.

$$\begin{aligned}&(\forall\varepsilon > 0)(\exists\delta > 0)(\forall\mathbf{z}(t))(\forall t \in [t_0,\infty))\left(\|\mathbf{x}(t_0) - \mathbf{z}(t_0)\| < \delta \Rightarrow \right.\\ &\left.\Rightarrow \|\mathbf{x}(t) - \mathbf{z}(t)\| < \varepsilon\right).\end{aligned} \tag{П3.22}$$

Иными словами, решение $\mathbf{x}(t)$ называется устойчивым, если другие достаточно близкие к нему в момент времени t_0 решения $\mathbf{z}(t)$ целиком находятся в узкой ε-трубке, построенной вокруг $\mathbf{x}(t)$.

Определение 2. Решение $\mathbf{x}(t)$ системы (П3.21) называется *асимптотически устойчивым* по Ляпунову, если оно устойчиво и существует такое $\Delta > 0$, что все решения $\mathbf{z}(t)$, удовлетворяющие условию $\left(\left\|\mathbf{x}(t_0) - \mathbf{z}(t_0)\right\| < \Delta\right)$, обладают свойством

$$\lim_{t \to \infty} \left\|\mathbf{x}(t) - \mathbf{z}(t)\right\| = 0 .$$

В случае асимптотической устойчивости близкие решения не только остаются близкими друг к другу, но и неограниченно сближаются при возрастании t.

Для определения неустойчивого решения достаточно построить отрицание определения 1.

Для систем разностных уравнений

$$\mathbf{y}(n+1) = \mathbf{g}\left(n, \mathbf{y}(n)\right), \quad \mathbf{y}(n_0) = \mathbf{y}_0, \quad n \in [n_0, \infty) \tag{П3.23}$$

понятие устойчивости вводится аналогично предыдущему с заменой независимой переменной t на целую переменную n. Обозначим за $\mathbf{y}(n)$ и $\mathbf{w}(n)$ два решения (П3.23), отличающиеся начальными условиями $\mathbf{y}(n_0)$ и $\mathbf{w}(n_0)$.

Определение 3. Решение $\mathbf{y}(n)$ называется *устойчивым*, если для любого $\varepsilon > 0$ найдется такое $\delta > 0$, что для всех решений $\mathbf{w}(n)$ системы (П3.23), удовлетворяющих неравенству $\left(\left\|\mathbf{y}(n_0) - \mathbf{w}(n_0)\right\| < \delta\right)$, справедливо неравенство $\left(\left\|\mathbf{y}(n) - \mathbf{w}(n)\right\| < \varepsilon\right)$ при всех $n \in [n_0, \infty)$

$$\begin{gathered}(\forall \varepsilon > 0)(\exists \delta > 0)(\forall \mathbf{w}(n))(\forall n \in [n_0, \infty))\left(\left\|\mathbf{y}(n_0) - \mathbf{w}(n_0)\right\| < \delta \Rightarrow \right. \\ \left. \Rightarrow \left\|\mathbf{y}(n) - \mathbf{w}(n)\right\| < \varepsilon\right).\end{gathered} \tag{П3.24}$$

Понятие *асимптотической устойчивости* предлагается определить самостоятельно на основе определения 2.

Сформулированные определения позволяют сделать суждение об устойчивости после анализа уже полученных решений. С практической точки зрения важно судить об устойчивости, не решая задачу.

Это возможно, в частности, для линейных систем с постоянной матрицей (П3.13):

$$\frac{d\mathbf{x}(t)}{dt} = \mathbf{A}\mathbf{x}(t) + \mathbf{g}(t), \qquad \mathbf{x}(t_0) = \mathbf{x}_0.$$

Будем называть их устойчивыми (асимптотически устойчивыми, неустойчивыми), если все их решения устойчивы (асимптотически устойчивы, неустойчивы).

Пусть $\mathbf{x}(t)$ и $\mathbf{z}(t)$ — два различных решения (П3.13), отличающиеся начальными условиями. В соответствии с (П3.14) они имеют вид:

$$\mathbf{x}(t) = e^{\mathbf{A}t}\mathbf{x}_0 + \int_0^t e^{\mathbf{A}(t-\tau)}\mathbf{g}(\tau)d\tau = e^{\mathbf{A}t}\mathbf{x}_0 + \int_0^t e^{\mathbf{A}t}\mathbf{g}(t-\tau)d\tau,$$

$$\mathbf{z}(t) = e^{\mathbf{A}t}\mathbf{z}_0 + \int_0^t e^{\mathbf{A}(t-\tau)}\mathbf{g}(\tau)d\tau = e^{\mathbf{A}t}\mathbf{z}_0 + \int_0^t e^{\mathbf{A}t}\mathbf{g}(t-\tau)d\tau.$$

Вычтем из первой формулы вторую. После сокращения интегралов получаем

$$\mathbf{x}(t) - \mathbf{z}(t) = e^{\mathbf{A}t}\left(\mathbf{x}_0 - \mathbf{z}_0\right).$$

Пусть первоначально собственные значения матрицы $\mathbf{A}$ различны. Тогда, используя для матричной экспоненты формулу Лагранжа — Сильвестра (П3.10), имеем

$$\mathbf{x}(t) - \mathbf{z}(t) = e^{\mathbf{A}t}\left(\mathbf{x}_0 - \mathbf{z}_0\right) = \sum_{k=1}^{m} e^{\lambda_k t}\mathbf{S}_k(\mathbf{A}, \lambda_k)\left(\mathbf{x}_0 - \mathbf{z}_0\right).$$

Обращаясь к определениям 1 и 2, приходим к выводу о том, что для обеспечения неравенства $\left(\left\|\mathbf{x}(t) - \mathbf{z}(t)\right\| < \varepsilon\right)$ элементы матричной экспоненты при $t \to \infty$ должны быть ограничены. А это, в свою очередь, требует, чтобы вещественные части $\mathrm{Re}(\lambda_k)$ собственных значений были бы неположительные. Для асимптотической устойчивости условие $\lim\limits_{t\to\infty}\left\|\mathbf{x}(t) - \mathbf{z}(t)\right\| = 0$ выполняется, когда элементы матричной экспоненты при $t \to \infty$ стремятся к нулю, а вещественные части собственных значений, соответственно, отрицательные.

Если среди собственных значений есть кратные, условия несколько корректируются. Пусть, например, собственное значение λ_k имеет кратность s. Тогда в решении этой группе собственных значений отвечает слагаемое $P_{s-1}(t)e^{\lambda_k t}$, где $P_{s-1}(t)$ — полином степени $s-1$. Если для $\mathrm{Re}(\lambda_k) < 0$ асим-

птотическая устойчивость обеспечивается независимо от кратности корня ($P_{s-1}(t)e^{\lambda_k t} \to 0$ при $t \to \infty$), то при нулевой вещественной части $P_{s-1}(t) \to \pm\infty$ при $t \to \infty$ и условие $\left(\|\mathbf{x}(t) - \mathbf{z}(t)\| < \varepsilon\right)$ не выполняется.

Подведем итоги.

- Для асимптотической устойчивости необходимо и достаточно, чтобы для всех собственных значений выполнялись условия $\mathrm{Re}(\lambda_k) < 0$.
- Для устойчивости необходимо, чтобы $\mathrm{Re}(\lambda_k) \le 0$. При этом достаточно, чтобы среди собственных значений с нулевой вещественной частью не было бы кратных.
- Для неустойчивости необходимо наличие хотя бы одного собственного значения с $\mathrm{Re}(\lambda_k) > 0$ или кратных собственных значений с $\mathrm{Re}(\lambda_k) = 0$.

Теперь обратимся к системе разностных уравнений с постоянной матрицей:

$$\mathbf{y}(n+1) = \mathbf{B}\mathbf{y}(n) + \mathbf{g}(n).$$

Пусть $\mathbf{y}(n)$ и $\mathbf{w}(n)$ — ее два различных решения, отличающиеся начальными условиями. В соответствии с (П3.19) они имеют вид:

$$\mathbf{y}(n) = \mathbf{B}^n\mathbf{y}(0) + \sum_{k=0}^{n-1} \mathbf{B}^{n-k-1}\mathbf{g}(k) = \mathbf{B}^n\mathbf{y}(0) + \sum_{k=0}^{n-1} \mathbf{B}^k\mathbf{g}(n-k-1),$$

$$\mathbf{w}(n) = \mathbf{B}^n\mathbf{w}(0) + \sum_{k=0}^{n-1} \mathbf{B}^{n-k-1}\mathbf{g}(k) = \mathbf{B}^n\mathbf{w}(0) + \sum_{k=0}^{n-1} \mathbf{B}^k\mathbf{g}(n-k-1).$$

Вычитая из первой формулы вторую, после сокращения сумм получаем:

$$\mathbf{y}(n) - \mathbf{w}(n) = \mathbf{B}^n\left(\mathbf{y}_0 - \mathbf{w}_0\right).$$

Если все собственные значения матрицы $\mathbf{B}$ различны, то, воспользовавшись формулой Лагранжа — Сильвестра для $\mathbf{B}^n$, имеем:

$$\mathbf{y}(n) - \mathbf{w}(n) = \sum_{k=1}^{m} \mu_k^n \mathbf{S}_k(\mathbf{B}, \mu_k)\left(\mathbf{y}_0 - \mathbf{w}_0\right),$$

где μ_k — собственные значения матрицы $\mathbf{B}$. Аналогично предыдущему для обеспечения неравенства $\left(\|\mathbf{y}(n) - \mathbf{w}(n)\| < \varepsilon\right)$ элементы матрицы $\mathbf{B}^n$ при $n \to \infty$ должны быть ограничены. А это, в свою очередь, требует выполнения условий $|\mu_k| \le 1$ для всех собственных значений. Для асимптотической устой-

чивости неравенства должны быть строгими: $|\mu_k|<1$. Если собственное значение μ_k имеет кратность s, то, как и для дифференциальных уравнений, в решении появляется слагаемое $P_{s-1}(n)\cdot\mu_k^n$, где $P_{s-1}(n)$ — полином степени $s-1$. Для $|\mu_k|<1$ этот факт не оказывает влияния на условие устойчивости, но для $|\mu_k|=1$ условие устойчивости нарушается, если $P_{s-1}(n)\to\pm\infty$ при $n\to\infty$. Как результат, сформулируем условия устойчивости.

- Для асимптотической устойчивости необходимо и достаточно, чтобы для всех собственных значений выполнялись условия $|\mu_k|<1$.
- Для устойчивости необходимо, чтобы $|\mu_k|\le 1$. При этом достаточно, чтобы среди собственных значений с единичными модулями не было бы кратных.
- Для неустойчивости необходимо наличие хотя бы одного собственного значения с $|\mu_k|>1$ или кратных собственных значений с $|\mu_k|=1$.

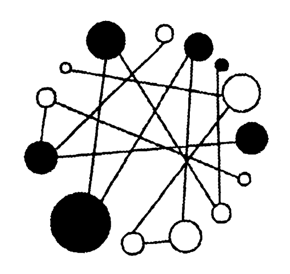
Глубже осознать понятие устойчивости позволяет ответ на следующий вопрос.

Вопрос

В какой взаимосвязи находятся понятия "устойчивости" и "ограниченности"? Имеются четыре варианта ответа.

- Из устойчивости решения следует его ограниченность.
- Из ограниченности решения следует его устойчивость.
- Оба предыдущих факта имеют место и по сути свойства ограниченности и устойчивости весьма близки.
- Это различные понятия. Из ограниченности решения еще не следует его устойчивость и из устойчивости не следует ограниченность.

ПРИЛОЖЕНИЕ 4



Степенные асимптотические разложения

Наиболее употребительным и известным степенным разложением функции $f(x)$ является ряд Тейлора. Это далеко не единственное степенное разложение, и для получения других используем общий подход, предложенный Ю. В. Ракитским, на основе метода неопределенных коэффициентов. Рассмотрим формулу Ньютона — Лейбница, в которой под интегралом выполнена замена переменной $\eta = t + h - \tau$:

$$x(t+h) = x(t) + \int_t^{t+h} \frac{dx(\eta)}{d\eta} d\eta = x(t) + \int_0^h \frac{dx(t+h-\tau)}{d(t+h-\tau)} d\tau.$$

Обозначение производной в последнем случае подчеркивает, что дифференцирование проводится по всему аргументу. Далее, внесем под интеграл множитель 1 и произведем интегрирование по частям:

$$x(t+h) = x(t) + \left(\int_0^\tau 1\, d\eta + C_1 \right) \frac{dx(t+h-\tau)}{d(t+h-\tau)} \Bigg|_0^h + \int_0^h (\tau + C_1) \frac{d^2 x(t+h-\tau)}{d(t+h-\tau)^2} d\tau =$$

$$= x(t) + \left(h + C_1\right) \frac{dx(t)}{dt} - C_1 \frac{dx(t+h)}{dt} + \int_0^h (\tau + C_1) \frac{d^2 x(t+h-\tau)}{d(t+h-\tau)^2} d\tau,$$

где C_1 — произвольная постоянная.

Продолжая интегрирование по частям s раз, каждый раз вводя новую постоянную, найдем общее степенное разложение:

$$x(t+h) = x(t) + \left(h + C_1\right)\frac{dx(t)}{dt} - C_1\frac{dx(t+h)}{dt} +$$
$$+\left(\frac{h^s}{s!} + C_1\frac{h^{s-1}}{(s-1)!} + \ldots + C_s\right)x^{(s)}(t) - \tag{П4.1}$$
$$- C_s x^{(s)}(t+h) + \int_0^h \left(\frac{\tau^s}{s!} + C_1\frac{\tau^{s-1}}{(s-1)!} + \ldots + C_s\right)\frac{d^{s+1}x(t+h-\tau)}{d(t+h-\tau)^{s+1}}d\tau.$$

Выбором произвольных постоянных C_k, $k = 1, 2, \ldots, s$ можно получить различные асимптотические разложения. В частности, если в (П4.1) выбрать все C_k нулевыми, то будем иметь хорошо известный явный ряд Тейлора.

$$x(t+h) = x(t) + h\frac{dx(t)}{dt} + \ldots + \frac{h^s}{s!}x^{(s)}(t) - \int_0^h \frac{\tau^s}{s!}\frac{d^{s+1}x(t+h-\tau)}{d(t+h-\tau)^{s+1}}d\tau. \tag{П4.2}$$

Определение C_k по формуле $C_k = \frac{(-h)^k}{k!}$ позволяет обратить в ноль все коэффициенты при производных в точке t:

$$\frac{h^s}{s!} + C_1\frac{h^{s-1}}{(s-1)!} + \ldots + C_s = \frac{h^s}{s!}\left(1 - s + \frac{s(s-1)}{2!} - \ldots + (-1)^s\right) = 0$$

и получить неявный ряд Тейлора:

$$x(t+h) = x(t) + h\frac{dx(t+h)}{dt} - \frac{h^2}{2}\frac{d^2x(t+h)}{dt^2} + \ldots - \frac{(-h)^s}{s!}x^{(s)}(t+h). \tag{П4.3}$$

Важное место среди степенных разложений занимает формула Эйлера — Маклорена, для построения которой необходим ряд сведений о числах и полиномах Бернулли.

Функция $G(t) = \frac{t}{\exp(t) - 1}$ называется производящей функцией для чисел Бернулли. Это означает, что коэффициенты B_k ее разложения в ряд будут числами Бернулли:

$$G(t) = \frac{t}{e^t - 1} = \sum_{k=0}^{\infty}\frac{B_k}{k!}t^k, \quad |t| < 2\pi.$$

Для их нахождения умножим последнее равенство на $(e^t - 1)$, представляя e^t в виде тейлоровского разложения:

$$t = \left(t + \frac{t^2}{2!} + \frac{t^3}{3!} + \ldots\right)\left(B_0 + B_1 t + \frac{B_2}{2!}t^2 + \frac{B_3}{3!}t^3 + \ldots\right).$$

Перемножая два ряда в правой части равенства и приравнивая коэффициенты при одинаковых степенях t, получаем формулы для нахождения B_k:

$$B_0 = 1, \quad \frac{B_0}{0!2!} + \frac{B_1}{1!1!} = 0,$$

$$\frac{B_0}{0!s!} + \frac{B_1}{1!(s-1)!} + \ldots + \frac{B_{s-1}}{(s-1)!1!} = 0, \quad s = 2,\ 3,\ \ldots$$

Умножая последнее равенство на $s!$ и прибавляя к обеим его частям по B_s, приходим к выражению

$$B_s = B_s + \binom{s}{1}B_{s-1} + \binom{s}{2}B_{s-2} + \ldots + \binom{s}{s-1}B_1 + B_0, \quad \binom{s}{k} = \frac{s!}{(s-k)!k!},$$

для запоминания которого используют мнемонический прием, основанный на сходстве этой формулы с биномом Ньютона $(B+1)^s$:

$$B_s = (B+1)^S_{\downarrow}.$$

Здесь подразумеваем формальное применение бинома Ньютона с перенесением показателя степени у B в индекс (B_S вместо B^S). Теперь без труда вычислим сами числа Бернулли:

$$B_0 = 1, \quad B_1 = -\frac{1}{2}, \quad B_2 = \frac{1}{6}, \quad B_3 = 0, \quad B_4 = -\frac{1}{30}, \quad B_5 = 0, \quad B_6 = \frac{1}{42},$$

$$B_7 = 0, \quad B_8 = -\frac{1}{30}, \quad B_9 = 0, \quad B_{10} = \frac{5}{66}$$

и т. д.

Нетрудно показать, что все нечетные числа Бернулли, начиная с третьего, равны нулю. Действительно,

$$G(-t) = \frac{-t}{\exp(-t) - 1} = \frac{t \cdot \exp(t)}{\exp(t) - 1} = t + G(t).$$

Раскладывая все функции в ряд, убеждаемся, что равенство $B_k(-t)^k = B_k t^k$ возможно лишь при $B_{2k+1} = 0,\ k = 1,\ 2,\ \ldots$

Для построения полиномов Бернулли $B_k(x)$ рассмотрим производящую функцию $H(t, x)$:

$$H(t,x) = \exp(xt)G(t) = \sum_{k=0}^{\infty} \frac{B_k(x)}{k!} t^k, \quad |t| < 2\pi .$$

Раскладывая $\exp(xt)$ и $G(t)$ в ряд и приравнивая коэффициенты при одинаковых степенях t^k, получим равенства

$$B_0(x) = 1, \quad \frac{B_1(x)}{1!} = \frac{B_0}{0!1!}x + \frac{B_1}{1!0!},$$

$$\frac{B_s(x)}{s!} = \frac{B_0}{0!s!}x^s + \ldots + \frac{B_{s-1}}{(s-1)!1!}x + \frac{B_s}{s!0!},$$

умножая которые на $s!$ приходим к формуле

$$B_s(x) = B_0 x^s + B_1 \binom{s}{1} x^{s-1} + \ldots + B_{s-1} \binom{s}{s-1} x + B_s .$$

И снова удобно воспользоваться мнемоническим приемом $B_s(x) = (x + B)_{\downarrow}^{S}$. Здесь индекс опускается только у чисел Бернулли B_k. Непосредственное вычисление позволяет записать:

$$B_0(x) = 1, \quad B_1(x) = x - \frac{1}{2}, \quad B_2(x) = x^2 - x + \frac{1}{6}, \ldots$$

При этом числа Бернулли — это соответствующие полиномы Бернулли в нулевой точке $B_k = B_k(0)$.

Теперь обратимся к построению формулы Эйлера — Маклорена, выбрав произвольные постоянные C_k таким образом, чтобы полином, стоящий под интегралом в (П4.1), был связан с полиномом Бернулли следующим образом:

$$\frac{\tau^s}{s!} + C_1 \frac{\tau^{s-1}}{(s-1)!} + \ldots + C_s = \frac{B_s(\tau / h) - B_s}{s!} h^s .$$

Параметры C_k находятся из сравнения полиномов в правой и левой частях. Раскрывая полином в правой части

$$h^s \frac{B_s(\tau/h)-B_s}{s!} = \frac{B_0\tau^s}{s!} + \frac{B_1 h\tau^{s-1}}{1!(s-1)!} + \ldots + \frac{B_{s-1}h^{s-1}\tau}{(s-1)!1!},$$

имеем

$$C_k = \frac{B_k h^k}{k!}, \quad k = 1,\ 2,\ \ldots,\ s-1; \quad C_s = 0.$$

Рассмотрим подробно коэффициенты при k-ой производной, предварительно подставив в них выражения для C_k:

$$\frac{h^k}{k!} + \frac{B_1 h}{1!}\frac{h^{k-1}}{(k-1)!} + \ldots + \frac{B_k h^k}{k!} = h^k\left(\frac{B_0}{0!k!} + \frac{B_1}{1!(k-1)!} + \ldots + \frac{B_{k-1}}{(k-1)!} + \frac{B_k}{k!}\right).$$

Сумма первых k членов в скобке равна нулю и при k-ой производной останется лишь множитель $B_k/k!$. Полином же при s-ой производной обращается в ноль, т. к. $C_s = 0$. С учетом этих рассуждений степенное разложение принимает вид:

$$x(t+h) - x(t) = h\frac{dx(t)}{dt} - \sum_{k=1}^{s-1}\frac{B_k h^k}{k!}\left(x^{(k)}(t+h) - x^{(k)}(t)\right) +$$
$$+\frac{h^s}{s!}\int_0^h \left(B_s(\tau/h) - B_s\right)\frac{d^{s+1}x(t+h-\tau)}{d(t+h-\tau)^{s+1}}d\tau.$$

Произведем замену переменных $\dfrac{dx(t+\tau)}{dt} = f(t+\tau)$. Тогда

$$x(t+h) - x(t) = \int_0^h f(t+\tau)d\tau,$$

а степенное разложение в новых обозначениях выглядит следующим образом:

$$\int_0^h \mathrm{f}(t+\tau)d\tau = hf(t) - \sum_{k=1}^{s-1}\frac{B_k h^k}{k!}\left[f^{(k-1)}(t+h) - f^{(k-1)}(t)\right] +$$
$$+\frac{h^s}{s!}\int_0^h \left(B_s(\tau/h) - B_s\right)\frac{d^s f(t+h-\tau)}{d(t+h-\tau)^s}d\tau \qquad (\text{П}4.4)$$

и называется *формулой Эйлера — Маклорена*. При $s=1$ и $s=2$ она превращается в малые квадратурные формулы левых прямоугольников и трапеций.

Возможно получение и составных квадратурных формул. С этой целью разобьем промежуток $[a, b]$ на N промежутков длиной h, для каждого из них применим формулу Эйлера — Маклорена и просуммируем результаты:

$$\int_a^b f(\tau)d\tau = h\sum_{i=0}^{N-1} f(t_i) - \sum_{i=0}^{N-1}\sum_{k=1}^{s-1}\frac{B_k h^k}{k!}\left[f^{(k-1)}(t_i+h) - f^{(k-1)}(t_i)\right] +$$
$$+\sum_{i=0}^{N-1}\frac{h^s}{s!}\int_0^h (B_s(\tau/h) - B_s) f^{(s)}(t_i + h - \tau)d\tau,$$

$$h = \frac{b-a}{N}.$$

Изменяя порядок суммирования, получаем:

$$\int_a^b f(\tau)d\tau = h\sum_{i=0}^{N-1} f(t_i) - \sum_{k=1}^{s-1}\frac{B_k h^k}{k!}\left[f^{(k-1)}(b) - f^{(k-1)}(a)\right] +$$
$$+\frac{h^s}{s!}\int_0^h (B_s(\tau/h) - B_s)\sum_{i=0}^{N-1} f^{(s)}(t_i + h - \tau)d\tau.$$

Последняя формула может быть использована не только как квадратурная, но и для вычисления сумм через интегралы. Выберем

$$f(\tau) = \tau^p, \quad h = 1, \quad \tau_0 = 0, \quad \tau_{N-1} = N-1, \quad s = p+1.$$

При таком значении s остаточный член обращается в ноль, тогда для суммы имеем:

$$\sum_{i=0}^{N-1} i^p = \frac{N^{p+1}}{p+1} + \frac{B_1}{1!}N^p + \frac{B_2}{2!}pN^{p-1} + \ldots + \frac{B_p p!}{p!}N = \sum_{k=1}^{p+1} N^k \frac{B_{p+1-k}p!}{(p+1-k)!k!}.$$

В качестве примера конкретизируем последнюю формулу для $p=3$:

$$\sum_{i=0}^{N-1} i^3 = N\frac{B_3 3!}{3!1!} + N^2\frac{B_2 3!}{2!2!} + N^3\frac{B_1 3!}{1!3!} + N^4\frac{B_0 3!}{0!4!} = \frac{N^2}{4} - \frac{N^3}{2} + \frac{N^4}{4} = \frac{N^2(N-1)^2}{4}.$$

Исключительный интерес представляет формула Дарбу — Обрешкова, получаемая из общего разложения (П4.1) путем выбора коэффициентов C_k на основе полиномов Лежандра:

$$\frac{\tau^s}{s!}+C_1\frac{\tau^{s-1}}{(s-1)!}+\ldots+C_s=\frac{1}{(2s)!}\frac{d^s}{d\tau^s}\left(\tau^2-h\tau\right)^s.$$

Разложение в этом случае приобретает вид:

$$x(t+h)=x(t)+\sum_{k=1}^{s}\frac{h^k}{k!}\frac{s!}{(2s)!}\frac{(2s-k)!}{(s-k)!}\left[(-1)^{k-1}x^{(k)}(t+h)+x^{(k)}(t)\right]+ \\ +\frac{h^{2s+1}}{(2s)!}\int_0^1(\rho^2-\rho)^s x^{(2s+1)}(t+\rho h)d\rho. \qquad (П4.5)$$

Эта формула обладает рядом специфических особенностей. Во-первых, коэффициенты разложения зависят от числа учитываемых членов s, что не наблюдалось для (П4.2)—(П4.4). Во-вторых, число учитываемых членов разложения равно s, в то время как остаточный член имеет множителем h^{2s+1} и содержит производную порядка $(2s+1)$. Последний факт указывает на предельно высокую точность формулы Дарбу — Обрешкова.

В качестве иллюстрации рассмотрим, как приближается функция e^x разложениями (П4.2)—(П4.5) с $s=1$ и $s=2$.

Явный ряд Тейлора (П4.2):

$$s=1:\ e^x\approx1+x;\qquad s=2:\ e^x\approx1+x+\frac{x^2}{2}. \qquad (П4.6)$$

Неявный ряд Тейлора (П4.3):

$$s=1:\ e^x\approx\frac{1}{1-x};\qquad s=2:\ e^x\approx\frac{1}{1-x+\dfrac{x^2}{2}}. \qquad (П4.7)$$

Формулы Эйлера — Маклорена и Дарбу — Обрешкова, как нетрудно убедиться, для $s=2$ совпадают:

$$s=1:\ e^x\approx\frac{1+x/2}{1-x/2};\qquad s=2:\ e^x\approx\frac{1+x/2+x^2/12}{1-x/2+x^2/12} \qquad (П4.8)$$

и начинают различаться с $s=3$.

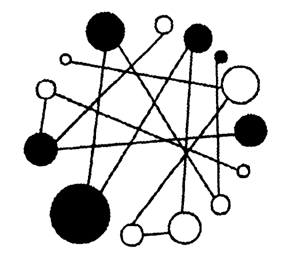
Формулы (П4.6)—(П4.8) хорошо известны при анализе устойчивости численных методов решения дифференциальных уравнений. Приближение (П4.6) характерно для явных методов Рунге — Кутты с известным ограничением на шаг интегрирования, что пагубно сказывается при решении жестких систем, а приближение (П4.7) — для неявных методов (см., например, неявный метод ломаных Эйлера). При больших по модулю отрицательных значениях x формула (П4.7) обеспечивает качественное сходство исходной функции и разложения, убывая с ростом $|x|$. Приближение (П4.8) для $s=1$ отвечает неявному методу трапеций, чья область устойчивости в точности равна левой полуплоскости. Этим же свойством обладают и методы, построенные на формуле (П4.8) с $s=2$. Ее погрешность в приближении экспоненты пропорциональна h^5.

В заключение рассмотрим следующую задачу. Пусть задан промежуток $[a,\ b]$. Для e^t по трем равноотстоящим узлам $t_0=a$, $t_1=(a+b)/2$, $t_2=b$ построим интерполяционный полином Лагранжа второй степени. Воспользуемся также для $\exp(t+h)$ приближениями по формулам (П4.2) и (П4.3) с $s=2$ и приближением (П4.5) с $s=1$ в точке $t=a$. Все четыре аппроксимирующие функции имеют в остаточном члене третью производную.

Вопрос

Какое из этих приближений лучше всего описывает исходную функцию $\exp(t+h)$ для $0<h<b-a$?

ПРИЛОЖЕНИЕ 5



Практические занятия

В рамках курса "Вычислительная математика" предусмотрены упражнения, лабораторные работы и курсовая работа. Примерное их содержание отражено в данном приложении.

П5.1. Упражнения

П5.1.1. Введение

Цель данной темы — выработать у студентов представление о том, что очень многие проблемы решаемых задач возникают по причине ограниченной точности представления чисел с плавающей точкой в компьютере, а также по причине погрешности задания исходных данных. Здесь весьма полезно рассмотреть ряд типовых примеров того, что сложение на компьютере может не обладать свойством ассоциативности. При этом в ряде случаев погрешность имеет недопустимо большую величину. Представляется полезным также решить следующие две хорошо известные задачи.

Задача 1 (более подробно см. [14], с. 29). Пусть требуется определить значение E_9, где E_n представляет собой следующий интеграл:

$$E_n = \int_0^1 x^n e^{x-1} dx .$$

Интегрируя по частям

$$E_n = \int_0^1 x^n e^{x-1} dx = x^n e^{x-1} \Big|_0^1 - \int_0^1 n x^{n-1} e^{x-1} dx ,$$

получаем рекуррентную формулу для определения E_9:

$$E_n = nE_{n-1}, \quad n = 2, 3, ..., 9. \tag{П5.1}$$

Задавая E_1 с точностью в шесть десятичных разрядов и выполняя в дальнейшем все вычисления по формуле (П5.1) без дополнительных округлений, получаем $E_9 \approx -0.0684800$. Так как вычисляемый интеграл положителен для любого *n*, ясно, что всего за восемь шагов погрешность стала недопустимо большой. В то же время, переписывая формулу (П5.1) в виде

$$E_{n-1} = \frac{1 - E_n}{n} \tag{П5.2}$$

и грубо полагая $E_{20} \approx 0$, после одиннадцати шагов получаем вполне удовлетворительный результат $E_9 \approx 0.0916123$. Этот пример — первое знакомство с *устойчивыми* и *неустойчивыми* алгоритмами. Перенос известного метода интегрирования по частям на компьютер в условиях ограниченной точности исходных данных дал негативный результат. Важно отметить, что замена алгоритма (П5.1) на любой устойчивый метод позволяет решить все возникшие проблемы.

Задача 2 (более подробно см. [49], с. 37). Требуется решить следующую систему линейных алгебраических уравнений $\mathbf{Ax} = \mathbf{b}$, где элементы матрицы $\mathbf{A}$ и вектора $\mathbf{b}$ заданы с предельной абсолютной погрешностью $\varepsilon \approx 0.005$ и принимают следующие значения:

$$\mathbf{A} = \begin{pmatrix} 1.00 & 0.99 \\ 0.99 & 0.98 \end{pmatrix}; \qquad \mathbf{b} = \begin{pmatrix} 1.99 \\ 1.97 \end{pmatrix}.$$

С одной стороны, точное решение этой системы имеет вид $\mathbf{x} = (1.00,\ 1.00)^T$. С другой стороны, если подставить в систему совсем другое решение $\mathbf{x} = (3.0000,\ -1.0203)^T$, то вектор $\mathbf{b}$ оказывается весьма близким к исходному $\mathbf{b} \approx (1.989903,\ 1.970106)^T$. Наконец, если в матрице $\mathbf{A}$ совсем немного изменить элемент $a_{22} = 0.98015$, то решение принимает следующий вид $\mathbf{x} = (3.97,\ -2.00)^T$. Здесь важно отметить, что в возникших проблемах "виновата" задача в своей постановке, ее решение очень чувствительно к погрешности в исходных данных, и смена метода не позволяет изменить ситуацию *(подробнее см. разд. 2.1)*.

П5.1.2. Погрешность арифметических операций

Эти вопросы, не отраженные в курсе лекций, необходимо подробно разобрать на упражнениях. Материал традиционно хорошо освещен в многочисленных учебниках (в частности, см. [1, 2]). Как результат, важно отметить, что основные неприятности связаны с операцией вычитания и, в первую очередь, с вычитанием близких чисел, когда происходит исчезновение верных разрядов числа. Среди многочисленных иллюстраций на эту тему могут быть следующие примеры.

Пример 1. Вычисление значения x по формуле $x = 7 - \sqrt{48.99}$ существенно лучше заменить на формулу $x = \dfrac{0.01}{7 + \sqrt{48.99}}$, где отсутствует вычитание близких чисел.

Пример 2. Требуется вычислить корни квадратного уравнения $x^2 + px + q = 0$ для следующих значений параметров: $p = 20\,000$, $q = 1$. При этом вычисления проводятся на компьютере с относительной погрешностью порядка $\varepsilon \sim 10^{-7}$. Вычисляя корни непосредственно по известным формулам

$$x_{1,2} = \frac{-p}{2} \pm \sqrt{\frac{p^2}{4} - q},$$

получаем явно ошибочное нулевое значение x_2:

$$x_1 = -10^4 - \sqrt{10^8 - 1} \approx -2 \cdot 10^4,$$

$$x_2 = -10^4 + \sqrt{10^8 - 1} \approx 0.$$

Здесь учитывался тот факт, что с учетом $\varepsilon \sim 10^{-7}$ выражение под знаком корня равно 10^8. Чтобы избежать вычитания близких чисел, значение x_1 будем вычислять, как и прежде, а значение x_2 с учетом того, что произведение корней равно q:

$$x_2 = \frac{q}{x_1} \approx 0.5 \cdot 10^{-4}.$$

Пример 3. Нужно вычислить значение $e^{-5.5}$, используя разложение экспоненты в ряд:

$$e^x = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \ldots$$

11 Зак. 3418

с относительной погрешностью промежуточных операций порядка $\varepsilon \sim 10^{-5}$ (более подробно см. [14], с. 27). Был получен ответ 0.0026363, в то время как истинный результат 0.00408677. В этом нет ничего удивительного, если заметить, что соседние слагаемые разного знака, наибольшее по модулю слагаемое равно –41.942, и абсолютная погрешность только этого слагаемого соизмерима с итоговым результатом. Здесь влияние операции вычитания не сразу бросается в глаза, т. к. верные знаки исчезают не сразу, а в течение нескольких последовательных операций. Избежать негативных эффектов можно, если воспользоваться тем же рядом для приближения экспоненты $e^{-5.5}$, когда операция вычитания отсутствует совсем, а затем применить очевидное равенство $e^{-5.5} = \frac{1}{e^{5.5}}$.

Пример 4. Вычисление средней точки в методе бисекции *(см. разд. 3.1)*. Пусть вычисления проводятся с тремя верными знаками мантиссы. $A = 0.891$, $B = 0.893$. Формула

$$C = (A + B)/2 = 1.78/2 = 0.89$$

дает результат вне промежутка $[A,\ B]$, и формула

$$C = A + (B - A)/2 = 0.891 + 0.002/2 = 0.892$$

оказывается более предпочтительной. В то же время для $A = -3.41$, $B = 8.73$ вторая формула

$$C = A + (B - A)/2 = -3.41 + 12.1/2 = -3.41 + 6.05 = 2.64$$

уступает по точности первой

$$C = (A + B)/2 = 2.66.$$

$$\textbf{if}(sign(A) \neq sign(B))\ \textbf{then}\ C = (A + B)/2\ \textbf{else}\ \ C = A + (B - A)/2\,.$$

П5.1.3. Конечные разности и суммирование функций

Предметом упражнений является теоретический материал, содержащийся в *разд. П1.1—П1.3*. При этом можно воспользоваться табл. П5.1 и П5.2.

Таблица П5.1. *Таблица разностей*

$f(k)$	$\Delta f(k)$
c	0
1	0
k	1
k^2	$2k+1$
2^k	2^k
a^k	$a^k(a-1)$
$(-a)^k$	$(-a)^k(a-1)$
a^{-k}	$a^{-k}(a^{-1}-1)$
$(-a)^{-k}$	$(-a)^{-k}((-a^{-1})-1)$
$1/k$	$-1/(k(k+1))$
$1/(k(k+1))$	$-2/(k(k+1)(k+2))$
$k^{[m]}$	$mk^{[m-1]}$
$k^{[-m]}$	$-mk^{[-m-1]}$
$\ln(k)$	$\ln((k+1)/k)$
$\sin(k\alpha)$	$2\cos((k+1/2)\alpha)\sin(\alpha/2)$
$\cos((k-1/2)\alpha)$	$-2\sin(k\alpha)\sin(\alpha/2)$
$\tan(k\alpha)$	$\sin(\alpha)/(\cos((k+1)\alpha)\cos(k\alpha))$
$\cot(k\alpha)$	$-\sin(\alpha)/(\sin((k+1)\alpha)\sin(k\alpha))$
$\mathrm{arctg}(k\alpha)$	$\mathrm{arctg}(\alpha/(1+\alpha^2 k(k+1)))$
$(-1)^k\sin((k+1/2)\alpha)$	$2(-1)^{k+1}\sin((k+1)\alpha)\cos(\alpha/2)$
$(-1)^k\cos((k-1/2)\alpha)$	$2(-1)^{k+1}\cos(k\alpha)\cos(\alpha/2)$

Таблица П5.2. *Таблица сумм*

$f(k)$	$\sum_{k=p}^{n-1} f(k)$
0	0
1	$n-p$
k	$(n^2-p^2-n+p)/2$
2^k	2^n-2^p
a^k	$(a^n-a^p)/(a-1)$
$(-a)^k$	$((-a)^n-(-a)^p)/(-a-1)$
a^{-k}	$(a^{-n}-a^{-p})/(a^{-1}-1)$
$(-a)^{-k}$	$((-a)^{-n}-(-a)^{-p})/((-a)^{-1}-1)$
$1/(k(k+1))$	$1/p-1/n$
$1/(k(k+1)(k+2))$	$-2(1/(n(n+1))-1/(p(p+1)))$
$k^{[m]}$	$(n^{[m+1]}-p^{[m+1]})/(m+1)$
$k^{[-m]}$	$((n-1)^{[1-m]}-(p-1)^{[1-m]})/(1-m)$
$\ln\left((k+1)/k\right)$	$\ln\left(n/p\right)$
$\cos((k+1/2)\alpha)$	$\sin\left((n-p)\alpha/2\right)\cos\left((n+p)\alpha/2\right)/\sin(\alpha/2)$
$\sin(k\alpha)$	$\sin\left((n-p)\alpha/2\right)\sin\left((n+p-1)\alpha/2\right)/\sin(\alpha/2)$
$1/(\cos((k+1)\alpha)\cos(k\alpha))$	$\sin((n-p)\alpha)/(\cos(n\alpha)\cos(p\alpha)\sin(\alpha))$
$1/\sin((k+1)\alpha)\sin(k\alpha)$	$\sin((n-p)\alpha)/(\sin(n\alpha)\sin(p\alpha)\sin(\alpha))$
$\operatorname{arctg}(\alpha/(1+\alpha^2 k(k+1)))$	$\operatorname{arctg}(\alpha(n-p)/(1+\alpha^2 np))$
$(-1)^{k+1}\sin((k+1)\alpha)$	$\dfrac{(-1)^n\sin((n+1/2)\alpha)-(-1)^p\sin((p+1/2)\alpha)}{2\cos(\alpha/2)}$
$(-1)^{k+1}\cos((k+1/2)\alpha)$	$\dfrac{(-1)^n\cos(n\alpha)-(-1)^p\cos(p\alpha)}{2\cos(\alpha/2)}$

В рамках этой темы рекомендуется также ознакомиться с методом неопределенных коэффициентов, например, на основе следующих примеров.

Пример 5. Вычислить значение суммы $\sum_{k=0}^{N}\cos(7k+4)$.

Первообразную $F(k)$, удовлетворяющую (П1.8), будем искать в виде синуса с произвольными коэффициентами $F(k)=A\sin(Bk+C)$:

$$\Delta F(k)=A\big(\sin(Bk+B+C)-\sin(Bk+C)\big)=$$
$$=2A\sin\frac{B}{2}\cos\left(Bk+\frac{B}{2}+C\right)=\cos(7k+4).$$

Отсюда $B=7$, $C=1/2$, $A=1/(2\cdot\sin(7/2))$. Значение $F(N+1)-F(0)$ и определяет значение искомой суммы.

Пример 6. Вычислить значение суммы $\sum_{k=0}^{N}k^2$.

Первообразную $F(k)$, удовлетворяющую (П1.8), будем искать в виде полинома третьей степени с произвольными коэффициентами $F(k)=Ak^3+Bk^2+Ck$:

$$\Delta F(k)=A\big((k+1)^3-k^3\big)+B\big((k+1)^2-k^2\big)+C=$$
$$=3Ak^2+(3A+2B)k+A+B+C=k^2.$$

Отсюда $A=\frac{1}{3}$, $B=-\frac{1}{2}$, $C=\frac{1}{6}$, $F(k)=\frac{(k-1)k(2k-1)}{6}$. Для искомой суммы в итоге имеем

$$\sum_{k=0}^{N}k^2=F(N+1)-F(0)=\frac{N(N+1)(2N+1)}{6}.$$

В качестве самостоятельного упражнения рекомендуется убедиться в справедливости следующего равенства:

$$\sum_{k=1}^{N}k^3=\left(\sum_{k=1}^{N}k\right)^2=\frac{n^2(n+1)^2}{4}.$$

П5.1.4. Линейное разностное уравнение порядка выше первого

Предметом упражнений является теоретический материал *разд. П1.4*, начиная с формулы (П1.23). Здесь важно подчеркнуть глубокую связь между дифференциальными и разностными уравнениями, и возможно параллельное решение этих уравнений как для однородного случая, так и для неоднородного. Далее приводится вариант домашнего расчетного задания на эту тему.

Расчетное задание.

1. Решить линейное дифференциальное уравнение второго порядка:

$$a_0 \frac{d^2x}{dt^2} + a_1 \frac{dx}{dt} + a_2 x = \varphi(t).$$

2. Решить линейное разностное уравнение второго порядка:

$$a_0 x_{n+2} + a_1 x_{n+1} + a_2 x_n = \varphi(n).$$

Частное решение неоднородного уравнения в обоих случаях найти двумя способами:

- подбором:
- методом Лагранжа вариации неопределенных коэффициентов.

Примечание. Коэффициенты a_0, a_1, a_2 и функция φ задаются преподавателем. В качестве φ рекомендуется задать полином первой степени от n или t соответственно. При этом для упрощения расчетов корни характеристического уравнения могут быть целыми (кратные или комплексные корни — на усмотрение преподавателя).

П5.1.5. Интерполяция функций

Построение полиномов Лагранжа и Ньютона может быть проиллюстрировано следующим расчетным заданием.

Расчетное задание.

Хорошо известная функция задана следующим фрагментом таблицы:

x	0	1/6	1/4	1/3
$f(x)$	0	1/2	$\sqrt{2}/2$	$\sqrt{3}/2$

Необходимо:

- угадать функцию;
- вычислить оценку остаточного члена для точек 1/12 и 5/24:
- построить интерполяционный полином в форме Лагранжа и в форме Ньютона;
- в точках 1/12 и 5/24 вычислить значение интерполяционного полинома и реальные погрешности интерполяции (с учетом того, что исходная функция известна) и сравнить их с оценкой остаточного члена;
- графически изобразить распределение оценки остаточного члена и реальной погрешности на всем промежутке.

Следует обратить внимание на последовательность построения "треугольных" таблиц после формул (1.5.4) и (1.5.5).

В начале *разд. 1.6* отмечалось, что на практике интерполяционные полиномы высоких степеней строят крайне редко. В первую очередь, это связано с тем, что их коэффициенты очень чувствительны к погрешностям исходных данных. Сравнительно малое изменение узлов интерполирования x_k или значений функции $f(x_k)$ приводит к сильному изменению вида самого полинома. Иллюстрацией этого являются следующие примеры.

Пример 7. Пусть в каждой точке таблицы с равноотстоящими узлами значения функции $f(x_k)$ вычислены с погрешностью ε (табл. П5.3). При этом в соседних точках эта погрешность имеет различный знак.

Таблица П5.3

x	$f(x)$	$\Delta f(x)$	$\Delta^2 f(x)$	$\Delta^3 f(x)$	$\Delta^4 f(x)$	$\Delta^5 f(x)$
x_0	$f_0+\varepsilon$	$\Delta f_0-2\varepsilon$	$\Delta^2 f_0+4\varepsilon$	$\Delta^3 f_0-8\varepsilon$	$\Delta^4 f_0+16\varepsilon$	$\Delta^5 f_0-32\varepsilon$
x_1	$f_1-\varepsilon$	$\Delta f_1+2\varepsilon$	$\Delta^2 f_1-4\varepsilon$	$\Delta^3 f_1+8\varepsilon$	$\Delta^4 f_1-16\varepsilon$	$\Delta^5 f_1+32\varepsilon$
x_2	$f_2+\varepsilon$	$\Delta f_2-2\varepsilon$	$\Delta^2 f_2+4\varepsilon$	$\Delta^3 f_2-8\varepsilon$	$\Delta^4 f_2+16\varepsilon$	$\Delta^5 f_2-32\varepsilon$
x_3	$f_3-\varepsilon$	$\Delta f_3+2\varepsilon$	$\Delta^2 f_3-4\varepsilon$	$\Delta^3 f_3+8\varepsilon$	$\Delta^4 f_3-16\varepsilon$	$\Delta^5 f_3+32\varepsilon$
x_4	$f_4+\varepsilon$	$\Delta f_4-2\varepsilon$	$\Delta^2 f_4+4\varepsilon$	$\Delta^3 f_4-8\varepsilon$	$\Delta^4 f_4+16\varepsilon$	$\Delta^5 f_4-32\varepsilon$
x_5	$f_5-\varepsilon$	$\Delta f_5+2\varepsilon$	$\Delta^2 f_5-4\varepsilon$	$\Delta^3 f_5+8\varepsilon$	$\Delta^4 f_5-16\varepsilon$	$\Delta^5 f_5+32\varepsilon$
...	...	...	...	...	...	...

На практике типичной является ситуация, когда с ростом порядка конечной разности k величина $\Delta^k f(x)$ убывает. Одновременно ее погрешность растет. Начиная с некоторого столбца, погрешность становится больше самой конечной разности, и дальнейшее увеличение степени интерполяционного полинома ни к чему хорошему не приводит. Этот пример может быть назван "излишне пессимистичным", т. к. погрешность во всех соседних точках имеет разный знак, что является наиболее неблагоприятным. Поэтому обратимся к "оптимистичному" примеру.

Пример 8. Здесь начальная погрешность имеется лишь в одной точке x_3 (табл. П5.4).

Таблица П5.4

x	$f(x)$	$\Delta f(x)$	$\Delta^2 f(x)$	$\Delta^3 f(x)$	$\Delta^4 f(x)$	$\Delta^5 f(x)$
x_0	f_0	Δf_0	$\Delta^2 f_0$	$\Delta^3 f_0 + \varepsilon$	$\Delta^4 f_0 - 4\varepsilon$	$\Delta^5 f_0 + 10\varepsilon$
x_1	f_1	Δf_1	$\Delta^2 f_1 + \varepsilon$	$\Delta^3 f_1 - 3\varepsilon$	$\Delta^4 f_1 + 6\varepsilon$	$\Delta^5 f_1 - 10\varepsilon$
x_2	f_2	$\Delta f_2 + \varepsilon$	$\Delta^2 f_2 - 2\varepsilon$	$\Delta^3 f_2 + 3\varepsilon$	$\Delta^4 f_2 - 4\varepsilon$	$\Delta^5 f_2 + 5\varepsilon$
x_3	$f_3 + \varepsilon$	$\Delta f_3 - \varepsilon$	$\Delta^2 f_3 + \varepsilon$	$\Delta^3 f_3 - \varepsilon$	$\Delta^4 f_3 + \varepsilon$	$\Delta^5 f_3 - \varepsilon$
x_4	f_4	Δf_4	$\Delta^2 f_4$	$\Delta^3 f_4$	$\Delta^4 f_4$	$\Delta^5 f_4$
x_5	f_5	Δf_5	$\Delta^2 f_5$	$\Delta^3 f_5$	$\Delta^4 f_5$	$\Delta^5 f_5$
...	...	...	...	...	...	...

И, хотя для пятой конечной разности вместо величины $\Delta^4 f_0 - 32\varepsilon$ получилось $\Delta^4 f_0 + 10\varepsilon$, погрешность неуклонно растет и распространяется по всем вышестоящим точкам.

Пример 9. Построить систему уравнений для определения коэффициентов кубического сплайна *(см. разд. 1.6)*. Исходная таблица имеет пять узлов. Здесь важно наглядно убедиться в том, что возникающая матрица системы является трехдиагональной.

Пример 10. По заданной таблице построить систему уравнений для определения коэффициентов интерполяционного полинома Эрмита (аналогично

(1.7.1)). Можно также отметить особый случай полинома Эрмита с одним узлом интерполирования (1.7.4).

П5.1.6. Численное дифференцирование и квадратурные формулы

Для иллюстрации идеи, лежащей в основе получения формул численного дифференцирования *(см. разд. 1.12)*, можно использовать следующий пример.

Пример 11. Построить формулу для второй производной по четырем равноотстоящим узлам таблицы.

При вычислении интеграла по составным квадратурным формулам в *разд. 1.9* был предложен следующий алгоритм.

1. Задаемся начальным значением N и вычисляем интеграл по любой составной квадратурной формуле, присваивая результат переменной I_{old}.
2. Удваиваем величину N и вновь вычисляем интеграл, присваивая результат переменной I_{new}.
3. Если значения I_{old} и I_{new} совпали с заданной точностью, то прекращаем работу. В противном случае присваиваем переменной I_{old} значение I_{new} и возвращаемся к шагу 2.

С использованием этой схемы может быть предложен следующий пример.

Пример 12. Написать стандартную программу, вычисляющую значение интеграла от произвольной функции по составной формуле Симпсона (1.9.20) с заданной точностью ε.

Общий подход к построению квадратурных формул на основе метода неопределенных коэффициентов и решение системы (1.10.2) отражают следующие примеры.

Пример 13. Для стандартного промежутка $[-1,\ 1]$ получить узлы и веса квадратурных формул Чебышева и Гаусса при $s=3$.

Пример 14. Квадратурные формулы Маркова характеризуются следующим свойством. Два узла равны пределам интегрирования ($x_1=a,\quad x_s=b$), а остальные параметры, как и в формулах Гаусса, получаются решением системы (1.10.2). Построить квадратурную формулу Маркова с тремя узлами. (Результатом является формула Симпсона.)

Представляется также полезным графически проиллюстрировать работу программы QUANC8 по выбору переменного шага интегрирования на примере функции на рис. 1.5.

П5.1.7. Среднеквадратичная аппроксимация и ортогональные полиномы

Для иллюстрации этого материала можно решать систему (1.13.3) как для непрерывно, так и для таблично заданной функции при $\varphi_k(x) = x^k$. В первом случае скалярные произведения записываются через интегралы, а во втором случае — через суммы. При обращении к ортогональным полиномам и процедуре Грама — Шмидта *(см. приложение 2)* вполне уместно изменить естественный порядок исходных линейно независимых функций $\varphi_k(x)$, например, на $1,\ x,\ x^2,\ x^3, \ldots$ Домашнее расчетное задание имеет следующий возможный вид.

Расчетное задание.

Дважды построить аппроксимирующий полином второй степени для функции $\frac{5}{x+6}$, $x \in [0,\ 1]$. Сначала воспользоваться методом наименьших квадратов в базисе $x,\ 1,\ x^2$ с непосредственным решением системы (2.13.3). Затем выполнить процедуру ортогонализации указанного базиса и построить полином уже в полученном ортогональном базисе. Графически отобразить погрешность аппроксимации.

П5.1.8. Задачи на матрицы. Векторно-матричное решение систем дифференциальных и разностных уравнений на основе формулы Лагранжа — Сильвестра

Выполнить следующие операции:

$$\mathbf{A} = \begin{pmatrix} 2 & -3 & 0 \end{pmatrix} \quad \mathbf{B} = \begin{pmatrix} 4 \\ 3 \\ 1 \end{pmatrix} \Rightarrow \mathbf{AB} = -1 \quad \mathbf{BA} = \begin{pmatrix} 8 & -12 & 0 \\ 6 & -9 & 0 \\ 2 & -3 & 0 \end{pmatrix};$$

$$\mathbf{A}=\begin{pmatrix}-2 & 3 & 0 & 1\\ 1 & 1 & 2 & -1\end{pmatrix}\quad \mathbf{B}=\begin{pmatrix}2 & 0\\ 1 & -1\\ -1 & 2\\ 1 & 3\end{pmatrix}\quad \mathbf{AB}=\begin{pmatrix}0 & 0\\ 0 & 0\end{pmatrix}\quad \mathbf{BA}=\begin{pmatrix}-4 & 6 & 0 & 2\\ -3 & 2 & -2 & 2\\ 4 & -1 & 4 & -3\\ 1 & 6 & 6 & -2\end{pmatrix}.$$

Решить матричные уравнения:

$$\begin{pmatrix}1 & 2\\ 3 & 4\end{pmatrix}\mathbf{X}=\begin{pmatrix}3 & 0\\ 7 & 2\end{pmatrix},\quad \mathbf{X}=\begin{pmatrix}1 & 2\\ 1 & -1\end{pmatrix};$$

$$\mathbf{X}\begin{pmatrix}1 & 2 & 3\\ 2 & 3 & 4\\ 3 & 4 & 1\end{pmatrix}=\begin{pmatrix}6 & 9 & 8\\ 0 & 1 & 6\end{pmatrix},\quad \mathbf{X}=\begin{pmatrix}1 & 1 & 1\\ 1 & 1 & -1\end{pmatrix}.$$

Найти все матрицы, коммутирующие с данными:

$$\mathbf{A}=\begin{pmatrix}1 & 2\\ 3 & 4\end{pmatrix},\quad \mathbf{AK}=\begin{pmatrix}2a & 2b\\ 3b & 2a+3b\end{pmatrix};$$

$$\mathbf{A}=\begin{pmatrix}1 & 0 & 0\\ 0 & 1 & 0\\ 3 & 1 & 2\end{pmatrix},\quad \mathbf{AK}=\begin{pmatrix}a & b & 0\\ c & d & 0\\ 3e-3a-c & e-3b-d & e\end{pmatrix}.$$

Найти собственные значения и собственные векторы заданной матрицы:

$$\begin{pmatrix}4 & -1 & -2\\ 2 & 1 & -2\\ 1 & -1 & 1\end{pmatrix},\quad \lambda_1=1,\quad \lambda_2=2,\quad \lambda_3=3;\quad \mathbf{u}_1=\begin{pmatrix}1\\ 1\\ 1\end{pmatrix},\quad \mathbf{u}_2=\begin{pmatrix}1\\ 0\\ 1\end{pmatrix},\quad \mathbf{u}_3=\begin{pmatrix}1\\ 1\\ 0\end{pmatrix}.$$

(Можно выполнить несколько примеров на иллюстрацию свойств матричной экспоненты из *разд. П3.6*.)

Далее представлены возможные варианты домашних расчетных заданий.

Расчетное задание.

Решить аналитически систему линейных дифференциальных уравнений:

$$\frac{d\mathbf{x}}{dt}=\mathbf{Ax}+\mathbf{b},\quad \mathbf{x}(0)=\mathbf{x}_0.$$

Решить аналитически систему линейных разностных уравнений:

$$\mathbf{x}_{n+1}=\mathbf{Ax}_n+\mathbf{b},\quad \mathbf{x}(0)=\mathbf{x}_0.$$

Необходимые матричные функции строить по формуле Лагранжа — Сильвестра. Матрица $\mathbf{A}$ и векторы $\mathbf{b}$, $\mathbf{x}_0$ задаются преподавателем.

Расчетное задание.

Для заданной матрицы $\mathbf{A}$ построить характеристическое уравнение и найти собственные значения, левые и правые собственные векторы. Для заданных векторов $\mathbf{b}$, $\mathbf{x}_0$ получить аналитическое решение системы

$$\frac{d\mathbf{x}}{dt} = \mathbf{A}\mathbf{x} + \mathbf{b}, \quad \mathbf{x}(0) = \mathbf{x}_0$$

в виде разложения по биортогональной системе собственных векторов.

П5.1.9. Решение систем нелинейных уравнений

Для решения системы нелинейных уравнений

$$\mathbf{f}(\mathbf{x}) = \mathbf{0}$$

воспользуемся методом Ньютона (3.2.3).

После предварительных упражнений на построение матрицы Якоби может быть предложено расчетное задание следующего вида.

Расчетное задание.

Решить систему нелинейных уравнений

$$\operatorname{tg}(x^{(1)}x^{(2)} + 0.4) = x^{(1)}x^{(1)};$$
$$0.6x^{(1)}x^{(1)} + 2x^{(2)}x^{(2)} = 1; \quad x^{(1)} > 0, \quad x^{(2)} > 0$$

методом Ньютона с абсолютной погрешностью $\varepsilon = 0.0001$. Начальное приближение найти графически.

Возможно использование различных модификаций метода Ньютона.

П5.1.10. Устойчивость численных методов решения систем обыкновенных дифференциальных уравнений

Эта тема является предметом лабораторного практикума, поэтому на практических занятиях можно ограничиться следующим иллюстративным примером.

Рассмотрим систему линейных однородных дифференциальных уравнений

$$\frac{d\mathbf{x}}{dt} = \mathbf{A}\mathbf{x}, \qquad t \in [0,\ 3]$$

с матрицей второго порядка, обладающей следующими собственными значениями: $\lambda_1 = -1$, $\lambda_2 = -20\ 000$. Будем решать эту систему численно явным методом ломаных Эйлера (4.0.3), который в данном случае приобретает вид

$$\mathbf{x}_{n+1} = \mathbf{x}_n + h\mathbf{A}\mathbf{x}_n = \left(\mathbf{E} + h\mathbf{A}\right)\mathbf{x}_n$$

или

$$\mathbf{x}_n = \left(\mathbf{E} + h\mathbf{A}\right)^n \mathbf{x}_0.$$

С другой стороны, точное решение исходной системы для дискретных значений $t_n = nh$ независимой переменной может быть записано следующим образом

$$\mathbf{x}(t_n) = e^{\mathbf{A}t_n}\mathbf{x}_0 = \left(e^{\mathbf{A}h}\right)^n \mathbf{x}_0.$$

Используя формулу Лагранжа — Сильвестра (см. (П3.16)), для точного и приближенного решения получаем соответственно

$$\mathbf{x}(t_n) = \left(e^{\lambda_1 h}\right)^n \mathbf{T}_1\mathbf{x}_0 + \left(e^{\lambda_2 h}\right)^n \mathbf{T}_2\mathbf{x}_0,$$

$$\mathbf{x}_n = \left(1 + h\lambda_1\right)^n \mathbf{T}_1\mathbf{x}_0 + \left(1 + h\lambda_2\right)^n \mathbf{T}_2\mathbf{x}_0,$$

где

$$\mathbf{T}_1 = \mathbf{u}_1\mathbf{v}_1^T, \qquad \mathbf{T}_2 = \mathbf{u}_2\mathbf{v}_2^T.$$

Легко заметить, что успех численного решения полностью определяется точностью аппроксимации экспоненты $e^{\lambda_k h}$ начальным отрезком ее ряда Тейлора $\left(1 + h\lambda_k\right)$. В остальном обе формулы совпадают.

Рассматриваемая система является жесткой. Исходя из условия устойчивости (4.3.2), шаг интегрирования должен строго удовлетворять неравенству $h < 10^{-4}$. Однако проведем следующий эксперимент. Выберем первоначально достаточно малый шаг интегрирования, а когда пограничный слой $\left(\tau_{\text{ПС}} < 10^{-3}\right)$ будет пройден и второе слагаемое в решении практически исчезнет, увеличим шаг до $h = 0.1$, что представляется достаточным для адек-

ватного описания первой экспоненты. Ясно, что условие устойчивости будет нарушено, но как в количественном плане будет вести себя решение?

Первое слагаемое в точном решении выглядит следующим образом:

$$\left(e^{\lambda_1 h}\right)^n = \left(e^{-0.1}\right)^n \approx (0.90484)^n,$$

а его аппроксимация в приближенном решении имеет вид:

$$\left(1 + h\lambda_1\right)^n = (0.9)^n,$$

весьма приемлемый для качественного представления решения.

Иная картина наблюдается для второго слагаемого. Для точного решения

$$\left(e^{\lambda_2 h}\right)^n = \left(e^{-2000}\right)^n$$

уже при $n = 1$ это слагаемое пренебрежимо мало, а для приближенного решения

$$\left(1 + h\lambda_2\right)^n = (-1999)^n$$

катастрофически быстро растет, увеличиваясь по модулю на одном шаге почти в 2000 раз! Даже если вне пограничного слоя задать начальные условия так, что $\mathbf{T}_2\mathbf{x}_0 = \mathbf{0}$, из-за погрешности метода и ограниченного представления в компьютере чисел с плавающей точкой возникнут малые ненулевые значения $\mathbf{T}_2\mathbf{x}_0$, которые будут катастрофически возрастать с каждым шагом ("взрыв погрешности").

А что в этом случае происходит с неявным методом ломаных Эйлера (4.0.4), рекомендованным для решения жестких систем? Для него имеем разностное уравнение

$$\mathbf{x}_{n+1} = \mathbf{x}_n + h\mathbf{A}\mathbf{x}_{n+1}, \quad \mathbf{x}_{n+1} = \left(\mathbf{E} - h\mathbf{A}\right)^{-1} \mathbf{x}_n$$

или

$$\mathbf{x}_n = \left(\mathbf{E} - h\mathbf{A}\right)^{-n} \mathbf{x}_0,$$

а использование формулы Лагранжа — Сильвестра приводит к выражению:

$$\mathbf{x}_n = \left(\frac{1}{1 - h\lambda_1}\right)^n \mathbf{T}_1\mathbf{x}_0 + \left(\frac{1}{1 - h\lambda_2}\right)^n \mathbf{T}_2\mathbf{x}_0.$$

Как и для явного метода, его первое слагаемое аппроксимирует первое слагаемое точного решения вполне приемлемо:

$$\left(\frac{1}{1-h\lambda_1}\right)^n = \left(\frac{1}{1.1}\right)^n \approx (0.9091)^n,$$

а второе слагаемое имеет следующий вид:

$$\left(\frac{1}{1-h\lambda_2}\right)^n = \left(\frac{1}{2001}\right)^n \approx (0.0005)^n.$$

Разумеется, эта величина никакого отношения не имеет к величине

$$\left(e^{\lambda_2 h}\right)^n = \left(e^{-2000}\right)^n,$$

но она быстро убывает с ростом n и не позволяет возрастать возникающей на шаге вычислительной погрешности.

П5.2. Лабораторные работы

Лабораторные работы являются иллюстрациями к отдельным разделам курса с использованием программ из книги Дж. Форсайта, М. Малькольма, К. Моулера "Машинные методы математических вычислений" [14], обсуждаемых в курсе лекций.

П5.2.1. Интерполяция и квадратурные формулы (программы *SPLINE*, *SEVAL*, *QUANC8*)

В некоторых заданиях использование SPLINE, SEVAL, с одной стороны, и QUANC8, с другой стороны, тесно взаимосвязано, а в некоторых — это относительно независимые части работы. Часто для изучения QUANC8 предлагаются интегралы, не имеющие конечного значения или содержащие подынтегральную функцию с разрывами, для которых программа совершенно не предназначена. Цель — изучить поведение программы в таких условиях.

Вариант 1. Для функции $f(x) = 1 - \exp(-x)$ по узлам $x_k = 0.3k$ ($k = 0, 1, ..., 10$) построить полином Лагранжа $L(x)$ 10-й степени и сплайн-функцию $S(x)$. Вычислить значения всех трех функций в точках $y_k = 0.15 + 0.3k$ ($k = 0, 1, ..., 9$).

Результаты отобразить графически. Используя программу QUANC8, вычислить интегралы:

$$\int_{0.5}^{1} (abs(\sin(x)\ -\ 0.6))^m dx \text{ , для } m=-1 \text{ и для } m=-0.5.$$

Вариант 2. Для функции $f(x)=1-\exp(-x)$ по узлам $x_k=0.3k$ ($k=0,\ 1, ..., 10$) построить полином Лагранжа $L(x)$ 10-й степени и сплайн-функцию $S(x)$. Затем сравнить значения трех интегралов:

$$\int_0^3 f(x)dx; \qquad \int_0^3 S(x)dx; \qquad \int_0^3 L(x)dx.$$

Первый интеграл вычислить аналитически, а для вычисления последнего использовать QUANC8.

Вариант 3. Для таблично заданной функции $f(x)$

x	0.0	0.2	0.5	0.7	1.0	1.3	1.7	2.0
$f(x)$	1.0	1.1487	1.4142	1.6245	2.0000	2.4623	3.2490	4.0000

построить сплайн-функцию и использовать ее для нахождения корня уравнения $f(x)+5x-3=0$ на промежутке $[0,\ 2]$ методом бисекции. (Программа метода бисекции создается самостоятельно.)

Вариант 4. Для $0 \le x \le 2$ с шагом $h=0.2$ вычислить значения функции $f(x)$ с использованием программы QUANC8, где $f(x)=\int_0^1 e^{xt}\sin(t)dt$. По полученным точкам построить сплайн-функцию и полином Лагранжа 10-й степени. Сравнить значения сплайн-функции и полинома с точным значением $f(x)$ (вычислить интеграл аналитически) в точках $x_k=(k-0.5)h$ для $k=1, 2, ..., 10$.

П5.2.2. Решение систем линейных алгебраических уравнений (программы *DECOMP* и *SOLVE*)

Здесь особое внимание уделяется исследованию зависимости погрешности решения от степени обусловленности матрицы.

Вариант 1. Матрица **X** строится следующим образом:

$$\mathbf{X}=\begin{pmatrix} x & 1 & 0 & 0 & \cdots & 0 \\ x^2 & x & 1 & 0 & \cdots & 0 \\ x^3 & x^2 & x & 1 & \cdots & 0 \\ x^4 & x^3 & x^2 & x & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & 1 \\ x^n & x^{n-1} & x^{n-2} & x^{n-3} & \cdots & x \end{pmatrix}.$$

Построить матрицы $\mathbf{A}=\beta\mathbf{E}+\mathbf{X}$ и вычислить матрицы $\mathbf{A}^{-1}$ для $x=2$ и $\beta=1.0,\ 0.1;\ 0.01;\ 0.001$ и т. д. вплоть до нештатной реакции компьютера. Размер матрицы **X** равен 5×5. В вычислительном эксперименте возможно изменение параметра x.

Вычислительный эксперимент состоит в исследовании связи априорной оценки "качества" заданных матриц, представленной оценкой стандартного числа обусловленности из программы DECOMP, и апостериорной, вычисляемой в ходе эксперимента: $\|\mathbf{R}\|=\left\|\mathbf{A}\mathbf{A}^{-1}-\mathbf{E}\right\|$, $\|\mathbf{A}\|=\max_i \sum_{j=1}^{n}\left|\grave{a}_{ij}\right|$.

Вариант 2. Матрица **A** формируется следующим образом: $\mathbf{A}=\beta\mathbf{E}+\mathbf{H}$, где

$$\mathbf{H}=\begin{pmatrix} \alpha & 1 & 0 & 0 & \cdots & 0 \\ \alpha^2 & \alpha & 1 & 0 & \cdots & 0 \\ \alpha^3 & \alpha^2 & \alpha & 1 & \cdots & 0 \\ \alpha^4 & \alpha^3 & \alpha^2 & \alpha & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & 1 \\ \alpha^n & \alpha^{n-1} & \alpha^{n-2} & \cdots & \cdots & \alpha \end{pmatrix}.$$

Решить линейные системы $\mathbf{A}\mathbf{x}_1=\mathbf{b}$, при $\beta=1.0,\ 0.1,\ 0.01$ и т. д. вплоть до нештатной реакции компьютера. $b_i=\dfrac{\sin(i)}{\cos(i)}$, $\alpha=2$. Размер матрицы $N=4$.

В вычислительном эксперименте возможно изменение параметра α. Сравнить полученные решения с решением систем $\mathbf{A}^T\mathbf{A}\mathbf{x}_2=\mathbf{A}^T\mathbf{b}$, полученных из исходных левой трансформацией Гаусса.

Априорная оценка качества заданных матриц представляется стандартным числом обусловленности из программы DECOMP, а апостериорная $\delta = \|x_1 - x_2\| / \|x_1\|$. Использовать сферическую норму вектора. Результаты сравнения представить в виде таблиц и графиков.

Вариант 3. Матрица $\mathbf{A} = \{a_{ij}\}$ формируется по следующему правилу:

$$a_{11} = 1, \quad (a_{ij} = j, \quad i \neq j), \quad (a_{ij} = \alpha + 1, \quad i = j \neq 1),$$

где α — параметр.

Исследовать связь априорной оценки и апостериорных оценок "качества" матриц при решении линейной системы $\mathbf{A}\mathbf{x}_1 = \mathbf{b}$ двумя способами. Первое получить с помощью программ DECOMP и SOLVE, а второе — по формуле $\mathbf{x}_2 = \mathbf{A}^{-1}\mathbf{b}$, где $\mathbf{A}^{-1}$ также вычисляется с помощью программ DECOMP и SOLVE. Априорной оценкой считать стандартное число программы DECOMP, а в качестве апостериорных взять относительную погрешность $\delta = \|x_1 - x_2\| / \|x_1\|$.

Размер матрицы $N = 7$; $\alpha = 0.9$; 0.99; 0.999 и т. д. вплоть до нештатной реакции компьютера, $b = (1, 1, 1, \ldots, 1)^T$. Возможно изменение компонентов вектора $\mathbf{b}$.

П5.2.3. Решение систем обыкновенных дифференциальных уравнений (программа *RKF45*)

Вариант работы. Решить систему дифференциальных уравнений:

$$\frac{dx^{(1)}}{dt} = -310x^{(1)} - 300x^{(2)} + \frac{1}{10t^2 + 1};$$

$$\frac{dx^{(2)}}{dt} = x^{(1)} + e^{-2t};$$

$$x^{(1)}(0) = 0; \quad x^{(2)}(0) = 1; \quad t \in [0, 0.4]$$

следующими способами с одним и тем же шагом печати $h_{\text{print}} = 0.02$:

- по программе RKF45 с $\varepsilon = 0.0001$;

- методом Рунге — Кутты 4-й степени точности

$$\mathbf{x}_{n+1} = \mathbf{x}_n + \frac{(\mathbf{k}_1 + 3\mathbf{k}_2 + 3\mathbf{k}_3 + \mathbf{k}_4)}{8};$$

$$\mathbf{k}_1 = h\mathbf{f}(t_n, \mathbf{x}_n); \quad \mathbf{k}_2 = h\mathbf{f}\left(t_n + \frac{h}{3}, \mathbf{x}_n + \frac{\mathbf{k}_1}{3}\right);$$

$$\mathbf{k}_3 = h\mathbf{f}\left(t_n + \frac{2h}{3}, \mathbf{x}_n - \frac{\mathbf{k}_1}{3} + \mathbf{k}_2\right); \quad \mathbf{k}_4 = h\mathbf{f}(t_n + h, \mathbf{x}_n + \mathbf{k}_1 - \mathbf{k}_2 + \mathbf{k}_3)$$

с двумя постоянными шагами интегрирования:

- $h_{\text{int}} = 0.01$;
- любой другой, позволяющий получить качественно верное решение.

Сравнить результаты. Определить максимально допустимый по устойчивости шаг интегрирования для заданного метода.

Примечание. Предлагаемый шаг $h_{\text{int}} = 0.01$ заведомо нарушает условие устойчивости.

П5.2.4. Проблема собственных значений и преобразования Хаусхолдера и Гивенса

Вариант 1. Написать стандартную программу, приводящую произвольную квадратную матрицу к форме Хессенберга на основе преобразования Хаусхолдера. Проверить ее на примере матрицы 6×6.

Вариант 2. Написать стандартную программу, приводящую произвольную квадратную матрицу к форме Хессенберга на основе преобразования Гивенса. Проверить ее на примере матрицы 6×6.

Вариант 3. Написать стандартную программу, осуществляющую QR-разложение произвольной квадратной матрицы. Проверить ее на примере матрицы 6×6.

Вариант 4. Написать стандартную программу, которая реализует QR-алгоритм для нахождения всех собственных значений заданной симметрической матрицы. (Используется программа, полученная в результате выполнения варианта 3.) Проверить ее на примере матрицы 6×6.

Вариант 5. Написать программу, находящую максимальное и минимальное по модулю собственные значения заданной матрицы с использованием прямого и обратного степенного метода. Проверить ее на примере матрицы 6×6.

П5.3. Курсовая работа

П5.3.1. Вычисление орбиты корабля "Аполлон"

Рассматривается движение капсулы "Аполлона" на орбите около Земли и Луны [14]. Начало координат находится в центре масс Луны и Земли, за ось x берется прямая, проходящая через эти два тела, а расстояние между ними принимается за единицу. Координатная система перемещается при вращении Луны вокруг Земли. Если μ — отношение массы Луны к массе Земли, то координаты Луны и Земли $(1-\mu,\ 0)$ и $(-\mu,\ 0)$ соответственно. Массой "Аполлона" пренебрегаем. $(x,\ y)$ — координаты "Аполлона".

$$r_1 = \left((x+\mu)^2 + y^2\right)^{\frac{1}{2}};\quad r_2 = \left((x-\mu^*)^2 + y^2\right)^{\frac{1}{2}};\quad \mu = \frac{1}{82.45};\qquad \mu^* = 1-\mu;$$

$$x'' = 2y' + x - \frac{\mu^*(x+\mu)}{r_1^3} - \frac{\mu(x-\mu^*)}{r_2^3};$$

$$y'' = -2x' + y - \frac{\mu^* y}{r_1^3} - \frac{\mu y}{r_2^3}.$$

Построить траекторию полета на промежутке $t \in [0,\ 20]$ с начальными условиями $x(0) = A$, $x'(0) = B$, $y(0) = C$, $y'(0) = D$.

Построить графики и интерпретировать их.

Оценить погрешность результата и влияние на точность траектории погрешности исходных данных. Значения A, B, C, D определяются следующим образом:

$$C = \left(\int_0^1 \frac{dt}{\sqrt{(t^2+1)(3t^2+4)}} - 0.40218305\right)^4;\quad D = -2.639594 \cdot x^*,$$

где x^* — наименьший корень уравнения: $2\lg x + 1 = \frac{x}{2}$; $A = 1.2$, $B = 0$.

П5.3.2. Решение краевой задачи методом стрельбы

Решить нелинейную краевую задачу относительно $y(x)$ на интервале $0 \le x \le 1$

$$\frac{d^2y}{dx^2} = y^2 - 1; \quad y(0) = 0; \quad y(1) = 1,$$

применив метод стрельбы, используя подпрограммы RKF45 и ZEROIN (рис. 4.3).

"Пристрелку" нужно вести по $y'(0)$ в диапазоне стрельбы $[A,\ B]$. Значения A и B задаются следующим образом:

$$B = 0.046 \int_{0.1}^{0.2} \frac{e^{-x}}{x^3} dx; \quad A = 1.2.$$

Оценить общую погрешность результата и влияние на точность результата погрешности исходных данных.

П5.3.3. Решение краевой задачи конечно-разностным методом с использованием метода Ньютона

Решить нелинейную краевую задачу относительно $y(x)$ на интервале $0 \le x \le 1$

$$\frac{d^2y}{dx^2} = y^2 - 1, \quad y(0) = Q, \quad y(1) = R$$

методом Ньютона, предварительно сведя исходное уравнение второго порядка к системе нелинейных алгебраических уравнений. С этой целью используется аппроксимация

$$\frac{d^2y}{dx^2}(x_i) \approx \frac{y(x_{i+1}) - 2y(x_i) + y(x_{i-1})}{h^2}, \quad x_i = x_0 + ih.$$

При этом система принимает вид:

$$y_{i+1} - 2y_i + y_{i-1} = h^2\left(y_i^2 - 1\right), \qquad i = 1,\ 2,\ \ldots,\ n-1,$$

$$y_0 = Q, \qquad y_n = R.$$

Решить эту систему для $n=10$ и $n=20$. Сравнить результаты. Оценить погрешность результата и влияние на ее величину погрешности исходных данных. При задании начального приближения для метода Ньютона приближенно считать, что функция $y(x)$ изменяется на промежутке $[0,\ 1]$ линейно.

Применение метода Ньютона к системе нелинейных алгебраических уравнений требует на каждой итерации решать систему линейных уравнений или обращать матрицу Якоби. Для этой цели использовать подпрограммы DECOMP и SOLVE.

Значения Q и R задаются формулами:

$$Q=\left(\int_0^{1/9}\sqrt{x}\cdot e^X dx-0.026392602\right)^4;$$

$R=1.559554\cdot z^*$, где z^* — значение z, минимизирующее функцию $f(z)=2z\left(1-0.5^z\right)+0.25^z+(z-1)^2$ на промежутке $[0.1;\ 1.0]$.

П5.3.4. Решение задачи параметрической идентификации (оценка параметров электрической цепи)

Электрическая цепь описывается следующей системой дифференциальных уравнений:

$$\frac{di_1}{dt}=\frac{1}{L_1}\left(E_1-E_2-U_C+i_3R_2-i_1\left(R_1+R_2\right)\right),$$

$$\frac{di_3}{dt}=\frac{1}{L_3}\left(E_2+U_C+i_1R_2-i_3\left(R_2+R_3\right)\right),$$

$$\frac{dU_C}{dt}=\frac{1}{C}\left(i_1-i_3\right).$$

Размыкание ключа происходит в момент времени $t=0$:

$$i_1(0)=\frac{E_1}{R_1},\quad i_3(0)=0,\quad U_C(0)=-E_2.$$

При этом $L_1 = L_3 = L$ и $R_1 = R_3 = R$. По заданной таблице экспериментальных данных оценить емкость конденсатора C с точностью до 0.0001 микрофарады. Воспользоваться подпрограммами `FMIN` и `RKF45`. Оценить точность результата и влияние на точность погрешности исходных данных. Значение C лежит в диапазоне от 0.5 до 2 микрофарад. В табл. П5.5 напряжение U_C дано в вольтах, а время — в миллисекундах.

Таблица П5.5

t	U_C **(Вольт)**
0	–1.000
0.1	7.777
0.2	12.017
0.3	10.701
0.4	5.407
0.5	–0.843
0.6	–5.159
0.7	–6.015
0.8	–3.668
0.9	0.283
1.0	3.829

Значения R, R_2, E_2 являются решением системы уравнений:

$$\begin{cases} 16R - 18R_2 + 24E_2 = 304 \\ -18R + 49R_2 - 42E_2 = 218 \\ 24R - 42R_2 + 46E_2 = 166. \end{cases}$$

Значения L, E_1 задаются следующими формулами:

$$L = 0.1469517 \cdot \int_0^1 \frac{\ln(1+x)}{1+x^2} dx;$$

$E_1 = 18.75217 \cdot x^*$, где x^* — корень уравнения: $e^x = 2(x-1)^2$.

Примечание. В рамках этой работы предполагается построение алгоритма по схеме, подобной рис. 4.3 для метода стрельбы с заменой программы ZEROIN на программу FMIN. Вызывающая программа MAIN обращается к FMIN для минимизации некоторой функции $\mathbf{F}(C)$ в диапазоне $C \in [C_{min}, C_{max}]$, где емкость C выступает в качестве единственного входного параметра. Значение функции $\mathbf{F}(C)$, которую реализует пользователь, на выходе определяется в соответствии с формулой

$$\mathbf{F}(C) = \sum_{k=0}^{10} \left(U_C(t_k) - U_C^{\text{эксп}}(t_k) \right)^2,$$

где t_k, $U_C^{\text{эксп}}(t_k)$ — экспериментальные данные таблицы, а $U_C(t_k)$ — значения решения системы дифференциальных уравнений при величине C, определяемой значением входного параметра. Для получения $U_C(t_k)$ программа $\mathbf{F}(C)$ вызывает процедуру RKF45, которая, в свою очередь, вызывает программу вычисления производных решения $\mathbf{f}(t, \mathbf{x})$.

Литература

Учебная литература ко всем разделам курса

1. Бахвалов Н. С., Жидков Н. П., Кобельков Г. М. Численные методы. — М.: Бином. Лаборатория знаний, 2007. — 640 с.
2. Березин И. С., Жидков Н. П. Методы вычислений // В 2 т. — Физматгиз, 1966. — Т. 1. — 632 с.
3. Березин И. С., Жидков Н. П. Методы вычислений // В 2 т. — Физматгиз, 1962. — Т. 2. — 620 с.
4. Волков Б. А. Численные методы. — СПб.: Лань, 2004. — 248 с.
5. Демидович Б. П., Марон И. А. Основы вычислительной математики. — СПб.: Лань, 2004. — 672 с.
6. Калиткин Н. Н. Численные методы. — М.: Наука, 1978. — 512 с.
7. Каханер Д., Моулер К., Нэш С. Численные методы и программное обеспечение. — М.: Мир, 2001. — 575 с.
8. Крылов В. И., Бобков В. В., Монастырный П. И. Вычислительные методы // В 2 т. — М.: Наука, 1976. — Т. 1. — 304 с.
9. Крылов В. И., Бобков В. В., Монастырный П. И. Вычислительные методы // В 2 т. — М.: Наука, 1977. — Т. 2. — 400 с.
10. Ланцош К. Практические методы прикладного анализа. — М.: Физматгиз, 1961. — 524 с.
11. Мак-Кракен Д., Дорн У. Численные методы и программирование на Фортране. — М.: Мир, 1977. — 582 с.
12. Самарский А. А. Введение в численные методы. — М.: Наука, 1997. — 239 с.
13. Самарский А. А., Гулин А. В. Численные методы. — М.: Наука. 1989. — 430 с.

14. Форсайт Дж., Малькольм М., Моулер К. Машинные методы математических вычислений. — М.: Мир, 1980. — 280 с.
15. Хемминг Р. В. Численные методы. — М.: Наука, 1972. — 400 с.

Литература к отдельным разделам курса

16. Арушанян О. Б., Залеткин С. В. Численное решение обыкновенных дифференциальных уравнений на Фортране. — М.: Изд-во МГУ, 1990. — 336 с.
17. Беллман Р. Введение в теорию матриц. — М.: Наука, 1976. — 351 с.
18. Васильев Ф. П. Численные методы решения экстремальных задач. — М.: Наука, 1988. — 520 с.
19. Воеводин В. В. Вычислительные основы линейной алгебры. — М.: Наука, 1977. — 304 с.
20. Воеводин В. В., Кузнецов Ю. А. Матрицы и вычисления. — М.: Наука, 1984. — 318 с.
21. Воеводин В. В., Воеводин Вл. В. Параллельные вычисления. — СПб.: БХВ-Петербург, 2004. — 608 с.
22. Гантмахер Ф. Р. Теория матриц. — М.: Наука, 2004. — 559 с.
23. Гельфонд А. О. Исчисление конечных разностей. — М.: УРСС, 2006. — 376 с.
24. Гилл Ф., Мюррей У., Райт М. Практическая оптимизация. — М.: Мир, 1985. — 509 с.
25. Голуб Дж., Ван Лоун Ч. Матричные Вычисления. — М.: Мир, 1999. — 548 с.
26. Джордж А., Лю Дж. Численное решение больших разреженных систем уравнений. — М.: Мир, 1984. — 336 с.
27. Дэнни Док., Шнабель Р. Численные методы безусловной оптимизации и решения нелинейных уравнений. — М.: Мир, 1988. — 440 с.
28. Зимницкий В. А., Устинов С. М. Методы анализа математических моделей динамических систем. — Л.: ЛГТУ, 1991. — 81 с.
29. Икрамов Х. Д. Несимметричная проблема собственных значений. — М.: Наука, 1991. — 240 с.

30. Икрамов Х. Д. Численное решение матричных уравнений. — М.: Наука, 1984. — 192 с.
31. Ланкастер П. Теория матриц. — М.: Наука, 1982. — 269 с.
32. Лоусон Ч., Хенсон Р. Численное решение задач метода наименьших квадратов. — М.: Наука, 1986. — 232 с.
33. Ортега Дж., Пул У. Введение в численные методы решения дифференциальных уравнений. — М.: Наука, 1986. — 288 с.
34. Ортега Дж., Рейнболдт В. Итерационные методы решения нелинейных систем уравнений со многими неизвестными. — М.: Мир, 1975. — 560 с.
35. Ортега Дж. Введение в параллельные и векторные методы решения линейных систем. — М.: Мир, 1991. — 368 с.
36. Парлетт Б. Симметричная проблема собственных значений. — М.: Мир, 1983. — 384 с.
37. Первозванский А. А. Поиск. — М.: Наука, 1970. — 264 с.
38. Писсанецки С. Технология разреженных матриц. — М.: Мир, 1988. — 410 с.
39. Райс Дж. Матричные вычисления и математическое обеспечение. — М.: Мир, 1984. — 264 с.
40. Ракитский Ю. В., Устинов С. М., Черноруцкий И. Г. Численные методы решения жестких систем. — М.: Наука, 1979. — 208 с.
41. Ракитский Ю. В., Устинов С. М., Сениченков Ю. Б., Воскобойников С. П. Алгоритмы и программы интегрирования дифференциальных уравнений // Учебн. пособие. — Л.: ЛПИ им. Калинина, 1982. — 88 с.
42. Самарский А. А. Теория разностных схем. — М.: Наука, 1983. — 616 с.
43. Самарский А. А., Гулин А. В. Численные методы математической физики. — М.: Научный мир, 2003. — 316 с.
44. Стренг Г. Линейная алгебра и ее применения. — М.: Мир, 1980. — 454 с.
45. Тихонов А. Н., Арсенин В. Я. Методы решения некорректных задач. — М.: Наука, 1986. — 288 с.
46. Уилкинсон Дж. Х. Алгебраическая проблема собственных значений. — М.: Наука, 1970. — 564 с.
47. Уилкинсон Дж., Райнш К. Справочник алгоритмов на языке АЛГОЛ. Линейная алгебра. — М.: Машиностроение, 1976. — 390 с.

48. Фаддеев Д. К., Фаддеева В. Н. Вычислительные методы линейной алгебры. — М.: Физматгиз, 1963. — 734 с.

49. Форсайт Дж., Молер К. Численное решение систем линейных алгебраических уравнений. — М.: Мир, 1969. — 168 с.

50. Хайрер Э., Нерсетт С., Ваннер Г. Решение обыкновенных дифференциальных уравнений. Нежесткие задачи. — М.: Мир, 1990. — 512 с.

51. Хайрер Э., Ваннер Г. Решение обыкновенных дифференциальных уравнений. Жесткие и дифференциально-алгебраические задачи. — М.: Мир, 1999. — 685 с.

52. Холл Дж., Уатт Дж. Современные численные методы решения обыкновенных дифференциальных уравнений. — М.: Мир, 1979. — 312 с.

53. Хорн Р., Джонсон Ч. Матричный анализ. — М.: Мир, 1989. — 655 с.

54. Черноруцкий И. Г. Методы оптимизации в теории управления. — СПб.: Питер, 2004. — 256 с.

55. Эстербю О., Златев З. Прямые методы для разреженных матриц. — М.: Мир, 1987. — 299 с.

Предметный указатель

B, L

Q

А

Б

В

Д

Е

З

И

К

Л

М

Н

О

П

Р

С

Т

У

Ф

Х

Ч

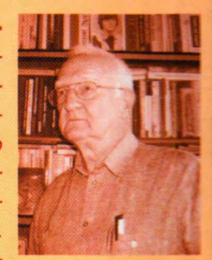
Ш

Э

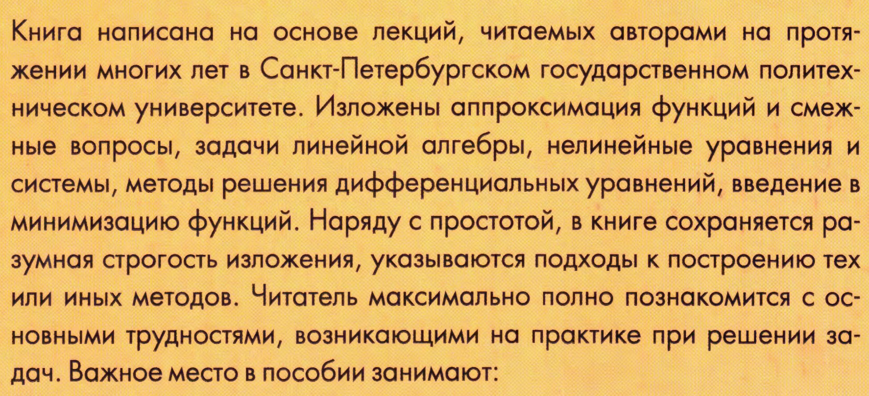


Устинов Сергей Михайлович, д. т. н., профессор кафедры «Информационные и управляющие системы» факультета технической кибернетики Санкт-Петербургского государственного политехнического университета. Имеет 35-летний стаж научно-педагогической деятельности. Автор свыше ста печатных работ.

Зимницкий Виктор Александрович, доцент кафедры «Информационные и управляющие системы» факультета технической кибернетики Санкт-Петербургского государственного политехнического университета. Стаж научно-педагогической работы – более 45 лет. Автор свыше десяти научно-методических работ.



ВЫЧИСЛИТЕЛЬНАЯ МАТЕМАТИКА



Книга написана на основе лекций, читаемых авторами на протяжении многих лет в Санкт-Петербургском государственном политехническом университете. Изложены аппроксимация функций и смежные вопросы, задачи линейной алгебры, нелинейные уравнения и системы, методы решения дифференциальных уравнений, введение в минимизацию функций. Наряду с простотой, в книге сохраняется разумная строгость изложения, указываются подходы к построению тех или иных методов. Читатель максимально полно познакомится с основными трудностями, возникающими на практике при решении задач. Важное место в пособии занимают:

- проблема плохой обусловленности при решении линейных систем алгебраических уравнений,
- явление жесткости в дифференциальных уравнениях,
- явление овражности при минимизации функций.

Дается представление о том, как строится программное обеспечение для обсуждаемых методов.

Книга предназначена для студентов, аспирантов, преподавателей технических вузов и инженеров.

БХВ-Петербург

194354, Санкт-Петербург,
ул. Есенина, 5Б

E-mail: mail@bhv.ru
Internet: www.bhv.ru

Тел/факс: (812) 591-6243




ISBN 978-5-9775-0318-1
9 785977 503181

С. М. Устинов
В. А. Зимницкий
ВЫЧИСЛИТЕЛЬНАЯ МАТЕМАТИКА

